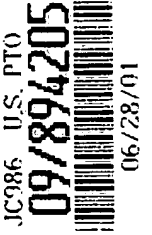


S/N unknown

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant: Kazushi KURATA Serial No.: unknown  
Filed: concurrent herewith Docket No.: 10873.753US01  
Title: DATA PROCESSING DEVICE AND PROGRAM CONVERSION DEVICE



CERTIFICATE UNDER 37 CFR 1.10

'Express Mail' mailing label number: EL815526673US

Date of Deposit: 28 June 2001

I hereby certify that this correspondence is being deposited with the United States Postal Service 'Express Mail Post Office To Addressee' service under 37 CFR 1.10 on the date indicated above and is addressed to Box Patents Application, Assistant Commissioner for Patents, Washington, D.C. 20231.

By: 

Name: Omesh Singh

SUBMISSION OF PRIORITY DOCUMENT(S)

Box Patents Application  
Assistant Commissioner for Patents  
Washington, D.C. 20231

Dear Sir:

Applicants enclose herewith one certified copy of a Japanese application, Serial No. 2000-202674, filed 4 July 2000, the right of priority of which is claimed under 35 U.S.C. § 119.

Respectfully submitted,

MERCHANT & GOULD P.C.  
P.O. Box 2903  
Minneapolis, Minnesota 55402-0903  
(612) 332-5300

By: 

Douglas P. Mueller  
Reg. No. 30,300

Dated: 28 June 2001

DPM:hjh

日 本 国 特 許 庁  
JAPAN PATENT OFFICE

JC986 U.S. PTO  
09/894205  
06/28/01

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出 願 年 月 日

Date of Application:

2000年 7月 4日

出 願 番 号

Application Number:

特願2000-202674

出 願 人

Applicant(s):

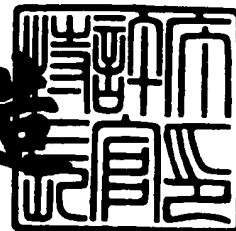
松下電器産業株式会社

CERTIFIED COPY OF  
PRIORITY DOCUMENT

2001年 5月31日

特 許 庁 長 官  
Commissioner,  
Japan Patent Office

及 川 耕 造



【書類名】 特許願

【整理番号】 R4137

【提出日】 平成12年 7月 4日

【あて先】 特許庁長官 殿

【国際特許分類】 G06F 15/78  
G06F 3/05

【発明者】

【住所又は居所】 大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内

【氏名】 蔵田 和司

【特許出願人】

【識別番号】 000005821

【氏名又は名称】 松下電器産業株式会社

【代理人】

【識別番号】 100095555

【弁理士】

【氏名又は名称】 池内 寛幸

【電話番号】 06-6361-9334

【選任した代理人】

【識別番号】 100076576

【弁理士】

【氏名又は名称】 佐藤 公博

【選任した代理人】

【識別番号】 100107641

【弁理士】

【氏名又は名称】 鎌田 耕一

【選任した代理人】

【識別番号】 100110397

【弁理士】

【氏名又は名称】 席丘 圭司

【選任した代理人】

【識別番号】 100115255

【弁理士】

【氏名又は名称】 辻丸 光一郎

【選任した代理人】

【識別番号】 100115152

【弁理士】

【氏名又は名称】 黒田 茂

【手数料の表示】

【予納台帳番号】 012162

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 0004605

【ブルーフの要否】 要

【書類名】 明細書

【発明の名称】 データ処理装置およびプログラム変換装置

【特許請求の範囲】

【請求項 1】 命令をシーケンスに従って順に読み込んで実行するデータ処理装置であって、

演算命令を読み込むフェッチ部と、

読み込んだ前記演算命令を解読する解読部と、

解読した前記演算命令を実行する実行部と、

前記演算命令の実行結果に従って、前記シーケンスにおいて前記演算命令に後続し前記演算命令に伴う一つまたは複数の後続命令のうちいずれか一つの後続命令以外の後続命令を無効化する命令無効化制御回路を備えたことを特徴とするデータ処理装置。

【請求項 2】 前記後続命令のうちの一つまたは複数の後続命令が複数の命令列からなる後続命令群である請求項 1 に記載のデータ処理装置。

【請求項 3】 記述プログラムを読み込んで機械語プログラムに変換するプログラム変換装置であって、

入力された記述プログラムの命令列から条件・判断を伴う命令を検出する命令検出手段と、

前記命令検出手段によって検出された条件・判断命令を、演算命令と、該演算命令に後続し該演算命令に伴う一つまたは複数の後続命令との命令列に書き換える命令書き換え手段を備えたことを特徴とするプログラム変換装置。

【請求項 4】 前記後続命令のうちの一つまたは複数の後続命令が複数の命令列からなる後続命令群である請求項 3 に記載のプログラム変換装置。

【請求項 5】 命令をシーケンスに従って順に読み込んで実行するデータ処理装置であって、

命令メモリとフェッチ部と解読部と命令実行部と命令無効化制御部とを備え、

前記命令メモリに、第 1 の条件付き命令から当該第 1 の条件の記述を除いた第 1 の後続命令と、第 2 の条件付き命令から当該第 2 の条件の記述を除いた第 2 の後続命令と、命令シーケンス上、前記第 1 の後続命令および第 2 の後続命令の前

に配置され、前記第 1 の後続命令と第 2 の後続命令とが相反する実行条件を持つことおよび前記第 1 の条件と第 2 の条件から生成された相反実行条件とを明示する指示命令とを記憶し、

前記フェッチ部が、前記命令メモリから前記指示命令をフェッチし、

前記解読部が、前記フェッチされた指示命令を解読し、

前記命令実行部が、前記解読した指示命令を実行し、

前記命令無効化制御部が、前記指示命令の実行結果より前記第 1 の条件と第 2 の条件のいずれが成立しているかを検知し、前記第 1 の後続命令と第 2 の後続命令のうち条件が成立していない一方の命令を無効化することを特徴とするデータ処理装置。

【請求項 6】 前記第 1 の条件付き命令、前記第 2 の条件付き命令、前記第 1 の後続命令、前記第 2 の後続命令のそれぞれが、一つの命令であるかまたは複数の命令列からなる命令群である請求項 5 に記載のデータ処理装置。

【請求項 7】 記述プログラムを読み込んで機械語プログラムに変換するプログラム変換装置であって、

入力された記述プログラムの命令列からプロセッサの状態を更新する可能性がある命令を検出する第 1 の命令検出手段と、

前記入力された命令列からプロセッサの状態に応じて実行するか否か決定される条件付きの命令を検出する第 2 の命令検出手段と、

前記第 1 の命令検出手段により検出された命令に後続して、前記第 2 の命令検出手段により検出された命令である第 1 の条件付き命令と第 2 の条件付き命令の 2 つの命令が連続しているかを判定する第 1 の判定手段と、

前記第 1 の条件付き命令と前記第 2 の条件付き命令の実行条件同士が互いに相反するものであるかを判定する第 2 の判定手段と、

機械語命令コードの中から、前記第 1 の命令検出手段により検出された命令と同一の処理実行を示し、かつ、後続する前記第 1 の条件付き命令および前記第 2 の条件付き命令が互いに相反する条件で実行されるものであることを示す指示命令を検索する指示命令検索手段と、

前記第 1 の命令検出手段と前記第 2 の命令検出手段によって検出された命令列

を、前記第 1 の条件付き命令から当該第 1 の条件記述を除いた第 1 の後続命令と、前記第 2 の条件付き命令から当該第 2 の条件記述を除いた第 2 の後続命令と、前記指示命令とに書き換える命令書き換え手段とを備えたことを特徴とするプログラム変換装置。

【請求項 8】 前記第 1 の条件付き命令、前記第 2 の条件付き命令、前記第 1 の後続命令、前記第 2 の後続命令のそれぞれが、一つの命令であるかまたは複数の命令列からなる命令群である請求項 7 に記載のプログラム変換装置。

【請求項 9】 命令をシーケンスに従って順に読み込んで実行するデータ処理装置であって、

演算命令を読み込むフェッチ部と、

読み込んだ前記演算命令を解読する解読部と、

解読した前記演算命令を実行する実行部と、

前記演算命令の実行結果に従って、前記シーケンスにおいて前記演算命令に後続し前記演算命令に伴う後続命令を無効化するか否か判断し、該判断に従って該後続命令を無効化する命令無効化制御回路を備えたことを特徴とするデータ処理装置。

【請求項 10】 前記後続命令が複数の命令列からなる後続命令群である請求項 9 に記載のデータ処理装置。

【請求項 11】 記述プログラムを読み込んで機械語プログラムに変換するプログラム変換装置であって、

入力された記述プログラムの命令列から条件・判断に伴う命令を検出する命令検出手段と、

前記命令検出手段によって検出された条件・判断命令を、演算命令と、該演算命令に後続し該演算命令に伴う後続命令の命令列に書き換える命令書き換え手段とを備えたことを特徴とするプログラム変換装置。

【請求項 12】 前記後続命令が複数の命令列からなる後続命令群である請求項 11 に記載のプログラム変換装置。

【請求項 13】 命令をシーケンスに従って読み込んで実行するデータ処理装置であって、

命令メモリとフェッチ部と解読部と命令実行部と命令無効化制御部とを備え、  
前記命令メモリが、条件付き命令から当該条件の記述を除いた後続命令と、命令シーケンス上、前記後続命令の前に配置され、前記後続命令が実行条件を持つことおよび前記実行条件を明示する後続実行条件指示命令とを記憶し、

前記フェッチ部が、前記命令メモリから前記後続実行条件指示命令をフェッチし、

前記解読部が、前記フェッチされた後続実行条件指示命令を解読し、

前記命令実行部が、前記解読した後続実行条件指示命令を実行し、

前記命令無効化制御部が、前記後続実行条件指示命令の実行結果より前記実行条件が成立しているかを検知し、前記実行条件が成立していない場合、前記後続命令を無効化することを特徴とするデータ処理装置。

【請求項 1 4】 前記条件付き命令、前記後続命令それぞれが、複数の命令列からなる命令群である請求項 1 3 に記載のデータ処理装置。

【請求項 1 5】 記述プログラムを読み込んで機械語プログラムに変換するプログラム変換装置であって、

入力された記述プログラムの命令列からプロセッサの状態を更新する可能性がある命令を検出する第 1 の命令検出手段と、

前記入力された命令列からプロセッサの状態に応じて実行するか否か決定される条件付きの命令を検出する第 2 の命令検出手段と、

前記第 1 の命令検出手段により検出された命令に後続して、前記第 2 の命令検出手段により検出された命令があるかを判定する第 1 の判定手段と、

機械語命令コードの中から、前記第 1 の命令検出手段により検出された命令と同一の処理実行を示し、かつ、後続の条件付き命令の実行条件を示す後続実行条件指示命令を検索する指示命令検索手段と、

前記第 1 の命令検出手段と第 2 の命令検出手段によって検出された命令列を、前記条件付き命令から当該条件記述を除いた条件隠し命令と前記後続実行条件指示命令に書き換える命令書き換え手段とを備えたことを特徴とするプログラム変換装置。

【請求項 1 6】 前記条件付き命令、前記後続命令のそれぞれが、複数の命令



列からなる命令群である請求項 1 5 に記載のプログラム変換装置。

【請求項 1 7】 前記命令無効化制御部が、前記フェッチ部に対して無効化した前記後続命令の読み込みをスキップさせる請求項 1、2、5、6、9、1 0、1 3、1 4 のいずれかに記載のデータ処理装置。

【発明の詳細な説明】

【0 0 0 1】

【発明の属する技術分野】

本発明は、条件付きで命令を実行するデータ処理装置に関し、特に、命令の割り付けに対する自由度の低下および命令コード長の増大に伴うプログラムメモリ容量の増加などの問題を発生することなく、パイプラインによる処理効率の高いデータ処理装置およびプログラム変換装置を提供する技術に関する。

【0 0 0 2】

【従来の技術】

データ処理は、科学技術計算や制御などその用途に関わらず、実行時に成立する条件に応じて処理を選択する場合が多い。図 3 2 に場合分け処理のフローチャートを示す。同図は、プロセッサの状態によって処理が選択される例であり、頻繁に出現するデータ処理である。まず、ステップ 1 0 0 1 で算術演算を実行し、次にステップ 1 0 0 2 で前ステップ 1 0 0 1 による演算結果がゼロか否かを判定する。その判定結果、ゼロでない場合にはステップ 1 0 0 4 に移り処理 2 を実行するが、ステップ 1 0 0 2 による判定結果がゼロの場合にはステップ 1 0 0 3 で処理 1 を実行した後、ステップ 1 0 0 5 に移る。

【0 0 0 3】

図 3 3 は、条件分岐命令によって図 3 2 の処理を実施するアセンブリ言語プログラムの例である。まず第 1 行目では減算を行う。同時にプロセッサは演算結果の性質を表すプロセッサステータスを生成する。例えば、プロセッサステータスには桁上がりが生じたことを示すキャリーフラグ、結果が正か負かを示すネガティブフラグ、結果が表現できるデータの範囲を超えたことを示すオーバフローフラグ、結果がゼロかを示すゼロフラグがある。次に第 2 行目では、ゼロフラグによる条件分岐命令が書かれており、もしゼロフラグが 1 なら、第 3 行目の命令 1

を実行し、第 4 行目で無条件に label2 に分岐して第 6 行目の命令 3 を実行する。もしゼロフラグが 0 なら label1 である第 5 行目に分岐することで第 3 行目の命令は実行せずにスキップして命令 2、命令 3 を順番に実行する。

## 【 0 0 0 4 】

上記従来技術のように、条件に応じて分岐する場合、処理性能を向上させる一般的な手法であるパイプライン処理を行っているプロセッサでは分岐によるパイプラインの乱れにより処理効率が低下するという問題がある。そこで分岐命令を削減するために条件付き実行命令が用いられることがある。

## 【 0 0 0 5 】

図 3 4 は、条件付き実行命令によって図 3 2 の処理フローを実施するアセンブリ言語プログラムである。アセンブラ的なニーモニックで記述している。第 1 行目の [SUB R0,R1] は、R0 レジスタの値から R1 のレジスタの値を減算して結果を R1 レジスタに格納する。このとき演算結果がゼロであればゼロフラグは 1 に、そうでなければ 0 になる。第 2 行目の [MOVIFZF R2,R4] は、ゼロフラグが 1 であれば、R2 レジスタの値を R4 レジスタに格納する。第 3 行目の [MOVIFNZF R3,R4] は、ゼロフラグが 0 であれば、R3 レジスタの値を R4 レジスタに格納する。第 4 行目の [ADD R5,R6] は R6 レジスタの値に R5 レジスタの値を加算して結果を R6 レジスタに格納する。ここで第 2 行目と第 3 行目の命令は条件付き実行命令である。図 3 5 に条件付き実行命令の命令フォーマットを示す。命令コード 4 1 と条件指定フィールド 4 0 で条件付き実行命令が表されている。例えば、条件指定フィールド 4 0 は、4 種類のステータスの真、偽を実行の条件とするなら 3 ビットで表されるものとなる。条件付き実行命令は条件指定フィールド 4 0 の条件と一致するときのみ命令を実行し、不一致するときは実行しない。ここで、常に無条件で実行されるのか、条件付き実行命令であるか、どの条件を使用するかという情報は、その個々の命令が解読された時点で判明する。そのため、プロセッサは結局実行されない命令であっても解読することとなる。

## 【 0 0 0 6 】

## 【発明が解決しようとする課題】

上記従来技術における条件付き実行命令によるデータ処理に関しては以下の間

題があった。

【0007】

ある条件に応じて処理を選択する場合、相反する条件付き実行命令が連続してプログラム中に配置されることが多く、それらは互いに排他的な関係にあることが多い。このようにある条件に応じて後続する処理のいずれか一方を実行し、他方を実行しない場合分けのプログラムはマイクロプロセッサ制御などあらゆる場面で広く見られる処理内容であるが、このとき必ず一方の条件付き実行命令は不成立となり実行されないものである。また、ある条件が成立するときのみ後続の一連の処理を実行し、当該条件が成立しないときには後続の一連の処理をスキップするという条件付き実行命令がプログラム中に配置されている場合、その条件が不成立となれば後続の一連の処理は実行されない。これらの場合、マイクロプロセッサ内ではパイプライン・フラッシュが発生することとなり、データ処理効率の低下を引き起こしていた。

【0008】

また、条件付き実行する処理に自由度をもたせるためには、数多くの種類からなる個々の命令を条件付き実行命令としてコード割付けし定義しておく必要があり、必然的にコード長を抑えることが難しくなってしまう。

【0009】

本発明は、係る問題点を鑑みてなされたものであり、その目的とするところは、条件付きで命令を実行するデータ処理装置に関して、命令の割り付けに対する自由度の低下および命令コード長の増大に伴うプログラムメモリ容量の増加の問題を発生することなく、処理効率の向上を図ることができるデータ処理装置およびプログラム変換装置を提供することにある。

【0010】

【課題を解決するための手段】

上記課題を解決するため、本発明の第1のデータ処理装置は、命令をシーケンスに従って順に読み込んで実行するデータ処理装置であって、演算命令を読み込むフェッチ部と、読み込んだ前記演算命令を解読する解読部と、解読した前記演算命令を実行する実行部と、前記演算命令の実行結果に従って、前記シーケンス

において前記演算命令に後続し前記演算命令に伴う一つまたは複数の後続命令のうちいずれか一つの後続命令以外の後続命令を無効化する命令無効化制御回路を備えたことを特徴とする。

## 【 0 0 1 1 】

上記構成により、先行する演算命令の演算結果に基づいて後続する命令のうち実行すべき1つの命令以外の命令を無効化することができ、実行しない命令の解読処理を行う必要がなく、処理効率の向上を図ることができる。また、個々の命令を条件付き実行命令としてコード割付けし定義しておく必要がなくなり、命令の割り付けに対する自由度の低下および命令コード長の増大に伴うプログラムメモリ容量の増加の問題を発生することがない。

## 【 0 0 1 2 】

上記本発明の第1のデータ処理装置において、前記後続命令のうちの一つまたは複数の後続命令が複数の命令列からなる後続命令群であることが好ましい。

## 【 0 0 1 3 】

上記構成により、後続命令のうちの幾つかが複数の命令が一群となったモジュールであっても本発明を適用することが可能となる。

## 【 0 0 1 4 】

また、上記課題を解決するため、本発明の第1のプログラム変換装置は、記述プログラムを読み込んで機械語プログラムに変換するプログラム変換装置であって、入力された記述プログラムの命令列から条件・判断に伴う命令を検出する命令検出手段と、前記命令検出手段によって検出された条件・判断命令を、演算命令と、該演算命令に後続し該演算命令に伴う一つまたは複数の後続命令との命令列に書き換える命令書き換え手段を備えたことを特徴とする。

## 【 0 0 1 5 】

上記構成により、普及しているプログラム記述により記述されたプログラムを書き換えて、本発明のデータ処理装置に適用する機械語プログラムを得ることができる。

## 【 0 0 1 6 】

上記本発明の第1のプログラム変換装置において、前記後続命令のうちの一つ

または複数の後続命令が複数の命令列からなる後続命令群であることが好ましい。

【 0 0 1 7 】

上記構成により、後続命令のうちの幾つかが複数の命令が一群となったモジュールとなる場合であっても本発明のデータ処理装置に用いる機械語プログラムを得ることができる。

【 0 0 1 8 】

また、本発明の第2のデータ処理装置は、命令をシーケンスに従って順に読み込んで実行するデータ処理装置であって、命令メモリとフェッチ部と解読部と命令実行部と命令無効化制御部とを備え、前記命令メモリに、第1の条件付き命令から当該第1の条件の記述を除いた第1の後続命令と、第2の条件付き命令から当該第2の条件の記述を除いた第2の後続命令と、命令シーケンス上、前記第1の後続命令および第2の後続命令の前に配置され、前記第1の後続命令と第2の後続命令とが相反する実行条件を持つことおよび前記第1の条件と第2の条件から生成された相反実行条件とを明示する指示命令とを記憶し、前記フェッチ部が、前記命令メモリから前記指示命令をフェッチし、前記解読部が、前記フェッチされた指示命令を解読し、前記命令実行部が、前記解読した指示命令を実行し、前記命令無効化制御部が、前記指示命令の実行結果より前記第1の条件と第2の条件のいずれが成立しているかを検知し、前記第1の後続命令と第2の後続命令のうち条件が成立していない一方の命令を無効化することを特徴とする。

【 0 0 1 9 】

上記構成により、指示命令の実行結果に基づいて、無効化制御部が後続する2命令の一方のみを実行し、他方を無効化するという排他制御を実現することができる。

【 0 0 2 0 】

上記本発明の第2のデータ処理装置において、前記第1の条件付き命令、前記第2の条件付き命令、前記第1の後続命令、前記第2の後続命令のそれぞれが、一つの命令であるかまたは複数の命令列からなる命令群であることが好ましい。

【 0 0 2 1 】

上記構成により、第 1 の条件付き命令、第 2 の条件付き命令、第 1 の後続命令、第 2 の後続命令が、一つの命令であったり複数の命令からなる命令群となったモジュールであったりなどいずれの組み合わせであっても本発明を適用することが可能となる。

## 【 0 0 2 2 】

また、本発明の第 2 のプログラム変換装置は、記述プログラムを読み込んで機械語プログラムに変換するプログラム変換装置であって、入力された記述プログラムの命令列からプロセッサの状態を更新する可能性がある命令を検出する第 1 の命令検出手段と、前記入力された命令列からプロセッサの状態に応じて実行するか否か決定される条件付きの命令を検出する第 2 の命令検出手段と、前記第 1 の命令検出手段により検出された命令に後続して、前記第 2 の命令検出手段により検出された命令である第 1 の条件付き命令と第 2 の条件付き命令の 2 つの命令が連続しているかを判定する第 1 の判定手段と、前記第 1 の条件付き命令と前記第 2 の条件付き命令の実行条件同士が互いに相反するものであるかを判定する第 2 の判定手段と、機械語命令コードの中から、前記第 1 の命令検出手段により検出された命令と同一の処理実行を示し、かつ、後続する前記第 1 の条件付き命令および前記第 2 の条件付き命令が互いに相反する条件で実行されるものであることを示す指示命令を検索する指示命令検索手段と、前記第 1 の命令検出手段と前記第 2 の命令検出手段によって検出された命令列を、前記第 1 の条件付き命令から当該第 1 の条件記述を除いた第 1 の後続命令と、前記第 2 の条件付き命令から当該第 2 の条件記述を除いた第 2 の後続命令と、前記指示命令とに書き換える命令書き換え手段とを備えたことを特徴とする。

## 【 0 0 2 3 】

上記構成により、普及しているプログラム記述により記述されたプログラムを、指示命令と後続する第 1 の後続命令と第 2 の後続命令に書き換えることができ、本発明のデータ処理装置に適用する機械語プログラムを得ることができる。

## 【 0 0 2 4 】

上記本発明の第 2 のプログラム変換装置において、前記第 1 の条件付き命令、前記第 2 の条件付き命令、前記第 1 の後続命令、前記第 2 の後続命令のそれぞれ

が、一つの命令であるかまたは複数の命令列からなる命令群であることが好ましい。

【 0 0 2 5 】

上記構成により、第 1 の条件付き命令、第 2 の条件付き命令、第 1 の後続命令、第 2 の後続命令が、一つの命令であったり一群となったモジュールとなる場合であったりなどいずれの組み合わせであっても本発明のデータ処理装置に用いる機械語プログラムを得ることができる。

【 0 0 2 6 】

次に、本発明の第 3 のデータ処理装置は、命令をシーケンスに従って順に読み込んで実行するデータ処理装置であって、演算命令を読み込むフェッチ部と、読み込んだ前記演算命令を解読する解読部と、解読した前記演算命令を実行する実行部と、前記演算命令の実行結果に従って、前記シーケンスにおいて前記演算命令に後続し前記演算命令に伴う後続命令を無効化するか否か判断し、該判断に従って該後続命令を無効化する命令無効化制御回路を備えたことを特徴とする。

【 0 0 2 7 】

上記構成により、先行する演算命令の演算結果に基づいて後続する命令を無効化するか否かを制御することができ、実行しない命令の解読処理を行う必要がなく、処理効率の向上を図ることができる。また、個々の命令を条件付き実行命令としてコード割付けし定義しておく必要がなくなり、命令の割り付けに対する自由度の低下および命令コード長の増大に伴うプログラムメモリ容量の増加の問題を発生することがない。

【 0 0 2 8 】

上記本発明の第 3 のデータ処理装置において、前記後続命令が複数の命令列からなる後続命令群であることが好ましい。

【 0 0 2 9 】

上記構成により、後続命令が複数の命令が一群となったモジュールであっても本発明を適用することが可能となる。

【 0 0 3 0 】

次に、本発明の第 3 のプログラム変換装置は、記述プログラムを読み込んで機

械語プログラムに変換するプログラム変換装置であって、入力された記述プログラムの命令列から条件・判断を伴う命令を検出する命令検出手段と、前記命令検出手段によって検出された条件・判断命令を、演算命令と、該演算命令に後続し該演算命令に伴う後続命令の命令列に書き換えることを特徴とする。

## 【 0 0 3 1 】

上記構成により、普及しているプログラム記述により記述されたプログラムを書き換えて、本発明の第 3 のデータ処理装置に適用する機械語プログラムを得ることができる。

## 【 0 0 3 2 】

上記本発明の第 3 のプログラム変換装置において、前記後続命令が複数の命令列からなる後続命令群であることが好ましい。

## 【 0 0 3 3 】

上記構成により、後続命令が複数の命令が一群となったモジュールとなる場合であっても本発明のデータ処理装置に用いる機械語プログラムを得ることができる。

## 【 0 0 3 4 】

次に、本発明の第 4 のデータ処理装置は、命令をシーケンスに従って読み込んで実行するデータ処理装置であって、命令メモリとフェッチ部と解読部と命令実行部と命令無効化制御部とを備え、前記命令メモリが、条件付き命令から当該条件の記述を除いた後続命令と、命令シーケンス上、前記後続命令の前に配置され、前記後続命令が実行条件を持つことおよび前記実行条件を明示する後続実行条件指示命令とを記憶し、前記フェッチ部が、前記命令メモリから前記後続実行条件指示命令をフェッチし、前記解読部が、前記フェッチされた後続実行条件指示命令を解読し、前記命令実行部が、前記解読した後続実行条件指示命令を実行し、前記命令無効化制御部が、前記後続実行条件指示命令の実行結果より前記実行条件が成立しているかを検知し、前記実行条件が成立していない場合、前記後続命令を無効化することを特徴とする。

## 【 0 0 3 5 】

上記構成により、指示命令の実行結果に基づいて、無効化制御部が後続命令を



実行するか無効化するという排他制御を実現することができる。

【 0 0 3 6 】

上記本発明の第 4 のデータ処理装置において、前記条件付き命令、前記後続命令それぞれが、複数の命令列からなる命令群であることが好ましい。

【 0 0 3 7 】

上記構成により、後続命令が複数の命令からなる命令群となったモジュールであっても本発明を適用することが可能となる。

【 0 0 3 8 】

次に、本発明の第 4 のプログラム変換装置は、記述プログラムを読み込んで機械語プログラムに変換するプログラム変換装置であって、入力された記述プログラムの命令列からプロセッサの状態を更新する可能性がある命令を検出する第 1 の命令検出手段と、前記入力された命令列からプロセッサの状態に応じて実行するか否か決定される条件付きの命令を検出する第 2 の命令検出手段と、前記第 1 の命令検出手段により検出された命令に後続して、前記第 2 の命令検出手段により検出された命令があるかを判定する第 1 の判定手段と、機械語命令コードの中から、前記第 1 の命令検出手段により検出された命令と同一の処理実行を示し、かつ、後続の条件付き命令の実行条件を示す後続実行条件指示命令を検索する指示命令検索手段と、前記第 1 の命令検出手段と第 2 の命令検出手段によって検出された命令列を、前記条件付き命令から当該条件記述を除いた条件隠し命令と前記後続実行条件指示命令に書き換える命令書き換え手段とを備えたことを特徴とする。

【 0 0 3 9 】

上記構成により、普及しているプログラム記述により記述されたプログラムを、指示命令と、当該指示命令の演算結果によりその実行、無効化が制御される後続命令に書き換えることができ、本発明のデータ処理装置に適用する機械語プログラムを得ることができる。

【 0 0 4 0 】

上記本発明の第 4 のプログラム変換装置において、前記条件付き命令、前記後続命令のそれぞれが、複数の命令列からなる命令群であることが好ましい。

【 0 0 4 1 】

上記構成により、後続命令が一群となったモジュールとなる場合であっても本発明のデータ処理装置に用いる機械語プログラムを得ることができる。

【 0 0 4 2 】

なお、上記本発明のデータ処理装置において、前記命令読み込み制御部が、前記フェッチ部に対して無効化した前記後続命令の読み込みをスキップさせることが好ましい。

【 0 0 4 3 】

上記構成により、命令無効化制御部の制御に基づいて、無効化された命令をスキップさせることができ、当該無効化命令が解読部に送られて解読処理が実行されないことでデータ処理装置の処理効率の低下を防止することができる。

【 0 0 4 4 】

【発明の実施の形態】

以下、本発明の実施形態について、図面を参照しながら説明する。

【 0 0 4 5 】

（実施形態 1）

実施形態 1 のデータ処理装置は、本発明の第 1 および第 2 のデータ処理装置の実施形態を示したものである。本実施形態 1 のデータ処理装置は、命令無効化制御回路を用いて、演算命令に後続する 2 つの命令のうちいずれか 1 つの後続命令を無効化するものである。

【 0 0 4 6 】

以下、本発明の実施の形態 1 について、図 1、図 2、図 3、図 4、図 5 を用いて説明する。なお、以下数値を “x' 00”、“x' 01” などと表すことがあるが、これは 16 進数による表記である。

【 0 0 4 7 】

図 1 はデータ処理装置の概略構成図である。データ処理装置 60 は、アセンブリ言語プログラム 61 を入力し機械語プログラム 63 に変換するプログラム変換装置 62 と、機械語プログラム 63 に基づいて動作するプロセッサ 64 とから構成されている。プロセッサ 64 は、命令フェッチステージ（以下、IF ステージと

略記する)、解読およびレジスタ読出しステージ(以下、DECステージと略記する)、演算およびメモリアクセスおよびレジスタ書き戻しからなる複数の処理を実行する実行ステージ(以下、EXステージと略記する)の3つのステージからなる3段パイプライン構造をもつものとする。

#### 【0048】

図2はプロセッサ64の構成を示すブロック図である。717は機械語プログラム63が格納されたROMである。706はROMに格納された命令をフェッチする命令フェッチ回路706である。714はフェッチした命令を格納しておく命令ラッチである。705は命令フェッチ回路706により取り込まれる命令を解読しプロセッサ64の各部を制御する命令解読器705である。715はパイプラインの制御を司るパイプライン制御信号である。708は命令の実行と無効化を制御し、パイプラインの動作を制御する命令無効化制御回路708である。709は条件付き実行の動作状態を表す条件付き実行ステータスである。707は命令を無効化する命令無効化信号である。716は条件付き実行ステータスを命令解読動作に応じて更新するシーケンサである。704はオペランドデータまたはオペランドを指定するためのアドレスを格納するレジスタファイル704であり、例えば、8本のレジスタR0からR7で構成されているとする。また、レジスタ長は32ビットであるものとする。703は算術論理演算を行う演算器である。712は算術論理演算の結果、反映されるプロセッサ状態を表すプロセッサステータスワード(PSW)である。702は本発明の指示命令によって先行的にプロセッサステータスの1つを選択し、命令無効化制御回路708に転送するためのセクタ702である。701はセクタ702により選択された排他的条件である。710、711は演算の対象となるデータをラッチするD1ラッチおよびD2ラッチである。713は命令の先読みを制御する命令先読み制御信号713である。

#### 【0049】

図3は、本発明の第1実施形態における命令無効化制御処理と条件付き実行ステータス遷移図である。まず、条件付き実行ステータス709の遷移について図3を用いて説明する。図3(a)は、命令無効化制御回路708がEXステージの

処理を無効化するか実行するかを表している（命令無効化信号 7 0 7）。同図の排他的条件 7 0 1 とは、本発明の指示命令によって選択されたプロセッサステータスの状態であり、0 か 1 の 2 値をとる。同図の条件付き実行ステータス 7 0 9 とは、同じく本発明の指示命令によって後続の命令の条件付き実行中の状態を表している。例えば、排他的条件 7 0 1 が “1” で条件付き実行ステータス 7 0 9 が “x'01” のときは命令無効化信号 7 0 7 を生成して、EXステージの処理を無効化する。ここで、命令無効化信号 7 0 7 は “1” のとき EXステージの処理を無効化し、“0” のとき EXステージの処理を実行するものとする。

## 【 0 0 5 0 】

図 3（b）は、条件付き実行ステータス 7 0 9 の遷移図であり、シーケンサ 7 1 6 により制御される。条件付き実行ステータス 7 0 9 は、指示命令が解読された時点で “x'00” から “x'11” にセットされ、パイプライン処理を進めるクロックに応じて、次の DECステージで後続の命令が解読され次の DECステージに移るとき “x'10” に更新される。さらに前記の条件付き実行ステータス 7 0 9 は、後続の命令を解読するに応じて、“x'10” から “x'01” に、“x'01” から “x'00” に更新される。つまり、このシーケンサ 7 1 6 は条件付き実行ステータス 7 0 9 を初期値 “x'00” から前記の指示命令が解読された後 “x'11” にセットし、次命令が解読されると “x'1” デクリメントし、最終的に “x'00” に遷移させる。

## 【 0 0 5 1 】

つまり、図 3（b）のように条件付き実行ステータスが “x'11”、“x'10”、“x'01”、“x'00” と順に遷移して行くことにより、排他的条件が “0” である場合には、条件付き実行ステータスが “x'10” にある第 2 行目の命令 2 が無効化されて実行されず、逆に排他的条件が “0” である場合には、条件付き実行ステータスが “x'01” にある第 3 行目の命令 3 の実行が無効化されて実行されないこととなり、排他的条件と条件付き実行ステータスに応じて命令 2 か命令 3 のいずれか一方が無効化されることとなる。

## 【 0 0 5 2 】

以上が本発明の第 1 および第 2 のデータ処理装置の構成例である。

## 【 0 0 5 3 】

以下に本発明の第 1 および第 2 のデータ処理装置の動作の例を説明する。一例として図 5 に示す機械語プログラム 1 を実行処理する場合の動作を説明する。図 4 は機械語プログラム 1 を表しており、ここでは説明の便宜上、ニーモニック表記している。その機械語プログラム 1 の各命令について簡単な説明を付する。

## 【 0 0 5 4 】

## (機械語プログラム 1)

第 1 行目：SUBIFEZF R0,R1 (R0レジスタの値からR1のレジスタの値を減算して結果をR1レジスタに格納する。さらに後続の 2 命令をプロセッサステータスのゼロフラグを条件として排他的な条件付き実行命令として指示する)

第 2 行目：MOV R2,R4 (R2レジスタの値をR4レジスタに格納する)

第 3 行目：MOV R3,R4 (R3レジスタの値をR4レジスタに格納する)

第 4 行目：ADD R5,R6 (R6レジスタの値にR5レジスタの値を加算して結果をR6レジスタに格納する)

上記の機械語プログラム 1 において、第 1 行目の指示命令SUBIFEZFは、後続する 2 つの命令の一方のみが実行されるという排他的な条件付き命令となっている。後述するようにSUBIFEZFの演算実行結果に基づいてゼロフラグの状態が決定され、当該ゼロフラグが示す排他的条件 7 0 1 の状態と条件付き実行ステータス 7 0 9 の状態に応じて、後続する第 2 行目の命令 2 と第 3 行目の命令 3 の一方のみが排他的に実行される。

## 【 0 0 5 5 】

図 5 は、プロセッサ 6 4 の動作タイミングを示す図である。図 5 は、図 4 の機械語プログラム 1 を実行する場合の動作を、パイプラインのIFステージ、DECステージ、EXステージで処理される命令をマシンサイクルと呼ばれるタイミング毎に示している。例えば、図 5 の命令 1 は、タイミング t 1 から t 3 までを順次パイプラインステージを経て処理される。このような命令の流れを、一般にパイプラインストリームと呼ぶ。また、図 5 に示すようにステージの前半（例えばタイミング t 1 1）と後半（タイミング t 1 2）を表すタイミングにより命令無効化制御回路 7 0 8 は動作するものとする。特にDECステージではタイミングの後半で命令無効化制御回路 7 0 8 により条件付き実行ステータス 7 0 9 を更新するも

のとする。またEXステージではタイミングの前半で命令無効化信号 7 0 7 に応じて、このステージで命令実行を開始するか、命令を無効化するか決定されるものとする。以下、時間が経過する順にタイミング毎にその動作を説明する。なお、レジスタ R0 と R1 には同値である “x' 0001” が格納され、R2 には “x' 0002”、R3 には “x' 0003” が格納されていたとする。

## 【 0 0 5 6 】

(タイミング t 1)

IFステージ：命令 1 [SUBIFEZF R0,R1]

タイミング t 1 1 : 命令 1 が R O M 7 1 7 から読出され、命令ラッチ 7 1 4 に格納される。

## 【 0 0 5 7 】

(タイミング t 2)

DECステージ：命令 1 [SUBIFEZF R0,R1]

タイミング t 2 1 : 命令ラッチ 7 1 4 に格納された命令 1 が命令解読器 7 0 5 で解読され、R0レジスタから値が読み出され D 1 ラッチ 7 1 0 に “x' 0001” が格納される。R1レジスタから値が読み出され D 2 ラッチ 7 1 1 に “x' 0001” が格納される。

## 【 0 0 5 8 】

タイミング t 2 2 : 命令無効化制御回路 7 0 8 は、条件付き実行ステータス 7 0 9 を “x' 00” から “x' 11” にセットする。排他的条件 7 0 1 の値を表わすレジスタとしてゼロフラグがセレクタ 7 0 2 により選択される。

## 【 0 0 5 9 】

IFステージ：命令 2 [MOV R2,R4]

タイミング t 2 1 : 命令 2 が R O M 7 1 7 から読み出され、命令ラッチ 7 1 4 に格納される。

## 【 0 0 6 0 】

(タイミング t 3)

EXステージ：命令 1 [SUBIFEZF R0,R1]

タイミング t 3 1 : 命令無効化信号 7 0 7 は初期状態のまま “0” となってお

り、EXステージの実行が行われ、命令1の実行が開始される。D1ラッチ710とD2ラッチ711に格納された値が演算器703で減算される。ここではD1ラッチ710とD2ラッチ711には同値“x'0001”が格納されているので、減算結果としてR1レジスタに“x'0000”が格納される。この結果ゼロフラグには“1”が格納され、排他的条件701が“0”から“1”となる。

## 【0061】

DECステージ：命令2 [MOV R2,R4]

タイミングt31：命令ラッチ714に格納された命令2が命令解読器705で解読され、R2レジスタから値が読出されD1ラッチ710に“x'0002”が格納される。

## 【0062】

タイミングt32：命令無効化制御回路708は、条件付き実行ステータス709を“x'11”から“x'10”に更新する。

## 【0063】

IFステージ：命令3 [MOV R3,R4]

タイミングt31：命令3がROM717から読出され、命令ラッチ714に格納される。

## 【0064】

(タイミングt4)

EXステージ：命令2 [MOV R2,R4]

タイミングt41：命令無効化信号707は“0”であるので、命令2の実行が開始される。D1ラッチ710に格納された値が読み出され演算器703をスルーしてR4レジスタに格納される。つまり、この例では後続命令のうち命令2が実行された。

## 【0065】

DECステージ：命令3 [MOV R3,R4]

タイミングt41：命令ラッチ714に格納された命令3が命令解読器705で解読され、R3レジスタから値が読出されD1ラッチ710に“x'0003”が格納される。

## 【 0 0 6 6 】

タイミング t 4 2 : 命令無効化制御回路 7 0 8 は、条件付き実行ステータス 7 0 9 を “x' 10” から “x' 01” に更新する。ここで、命令無効化制御回路 7 0 8 は、図 3 ( a ) に従って、条件付き実行ステータスが “x' 01” であり、排他的条件が “1” であるので命令無効化信号 7 0 7 を生成し、命令無効化信号 7 0 7 を “0” から “1” に更新する。

## 【 0 0 6 7 】

IFステージ：命令 4 [ADD R5,R6]

タイミング t 4 1 : 命令 4 が ROM 7 1 7 から読出され、命令ラッチ 7 1 4 に格納される。

## 【 0 0 6 8 】

(タイミング t 5 )

EXステージ：命令 3 [MOV R2,R4]

タイミング t 5 1 : 命令無効化信号 7 0 7 が “1” となっているEXステージは実行されず、命令 3 は無効化されて実行されない。つまり、この例では後続する命令のうち、命令 2 は実行され、命令 3 は無効化されて実行されない。

## 【 0 0 6 9 】

DECステージ：命令 4 [ADD R5,R6]

タイミング t 5 1 : 命令ラッチ 7 1 4 に格納された命令 4 が命令解読器 7 0 5 で解読され、R5レジスタから値が読出され D 1 ラッチ 7 1 0 に “x' 0005” が格納される。R6レジスタから値が読み出され D 2 ラッチ 7 1 1 に “x' 0006” が格納される。

## 【 0 0 7 0 】

タイミング t 5 2 : 命令無効化制御回路 7 0 8 は、条件付き実行ステータス 7 0 9 を “x' 01” から “x' 00” に更新する。ここで、命令無効化制御回路 7 0 8 は図 3 ( a ) に従って、条件付き実行ステータスが “x' 00” であり、排他的条件が “1” であるので、命令無効化信号 7 0 7 を “1” から “0” に更新する。

## 【 0 0 7 1 】

IFステージ：命令 5 [NOP]



タイミング t 5 1 : 命令 5 が R O M 7 1 7 から読出され、命令ラッチ 7 1 4 に格納される。

【 0 0 7 2 】

(タイミング t 6 )

EXステージ：命令 4 [ADD R5,R6]

タイミング t 6 1 : 命令無効化信号 7 0 7 は “ 0 ” であるので、EXステージが実行されることとなり、命令 4 が実行される。D 1 ラッチ 7 1 0 と D 2 ラッチ 7 1 1 に格納された値が演算器 7 0 3 で加算された後、R1レジスタに加算値が格納される。

【 0 0 7 3 】

DECステージ：命令 5 [NOP]

タイミング t 6 1 : 命令ラッチ 7 1 4 に格納された命令 5 が命令解読器 7 0 5 で解読される。NOP命令なので何も実行されない。

【 0 0 7 4 】

タイミング t 6 2 : 命令無効化制御回路 7 0 8 は、条件付き実行ステータス 7 0 9 を “ x'00 ” で保持する。

【 0 0 7 5 】

IFステージ：命令 6 [NOP]

タイミング t 6 1 : 命令 6 が R O M 7 1 7 から読出され、命令ラッチ 7 1 4 に格納される。

【 0 0 7 6 】

以上のように本実施形態 1 によれば、命令 1 [SUBIFEZF R0,R1] により排他的条件 7 0 1 としてゼロフラグを選択し、条件付き実行ステータス 7 0 9 をセットすることで後続の命令が特別な条件指定フィールドを付加することなく条件付き実行される。さらに後続の命令 2 と命令 3 を排他的な条件付き実行命令として指示することで、分岐によるパイプラインの乱れを生じることなく場合分けの処理が実現できる。

【 0 0 7 7 】

なお、上記の説明ではR0レジスタとR1レジスタの値を同値として、ゼロフラグ

の値が“1”となる場合を示したが、R0レジスタとR1レジスタの値が同値ではなく“0”となる場合も、命令無効化制御回路708は図3に従って命令無効化信号707を制御することにより命令無効化制御を実行することができる。この場合は後続命令のうち命令2が無効化されて実行されず、命令3が実行される。

【0078】

また、上記に示した例では、命令1はプロセッサステータスワード712のうちゼロフラグを命令無効化信号707として選択する命令としたが、限定されるものではなく、他のプロセッサステータスワード712を選択しても良い。

【0079】

(実施形態2)

実施形態2として、本発明の第1および第2のプログラム変換装置を図6、図7、図8、図9、図10を参照しつつ説明する。

【0080】

図6は、本発明のプログラム変換装置構成例である。201は、命令列を記憶する命令列記憶装置である。204は命令検出手段であり、第1の命令検出手段202および第2の命令検出手段203を備えている。第1の命令検出手段202は、入力された命令がプロセッサステータスを更新する命令かどうか検出するものである。第2の命令検出手段203は、入力された命令があるプロセッサの状態に応じて実行される条件付きの命令か否か検出するものである。208は、第1命令検出手段により検出されたある命令に対して、第2命令検出手段により検出された命令がその直後に配置されているか判定し記憶する第1判定手段である。209は、第1判定手段により記憶された命令の条件を判定する第2判定手段である。210は、第1命令検出手段による検出された命令と同一の処理を実行可能であって、かつ第2判定手段で判定された条件を明示する指示命令を検索する命令検索手段である。命令検索手段210では、プロセッサ64が実行できる全命令を記憶しているものとする。205は、命令列を書き換える命令書き換え手段である。なお、本実施形態2におけるプログラム変換装置は、206に示された第1の処理と207に示された第2の処理でアセンブリ言語プログラムを変換するものである。211は、第2の処理207の前半部である。

## 【 0 0 8 1 】

以上が、本発明の第 1 および第 2 のプログラム変換装置の構成例である。

## 【 0 0 8 2 】

次に、本実施形態 2 のプログラム変換装置によるプログラムの書き換え動作の例を説明する。図 7 は書き換え前のアセンブリ言語プログラム例であり、ニーモニック表記してある。図 8 は、本実施形態 2 のプログラム変換装置によって図 7 に示したプログラムを書き換えた機械語プログラムである。ただし、図 8 では、機械語プログラムの説明を簡単にするためにニーモニック表記してある。ここで、ニーモニック表記された両プログラムは、すでに前述したアセンブリ言語プログラムや機械語プログラムで説明されており、ADD 命令、MOV 命令などで使用しているオペランド（即値、レジスタ）が異なるだけであるのでここでは説明を省略する。

## 【 0 0 8 3 】

図 9 は、本発明の第 2 実施形態におけるプログラム変換装置による第 1 の処理を示す図である。この第 1 の処理 2 0 6 は、命令検出手段 2 0 4 により実行され、命令検出手段 2 0 4 により記述プログラムの命令列から条件・判断を伴う命令を検出することができる。命令検出手段 2 0 4 の処理は、第 1 命令検出手段 2 0 2 と第 2 命令検出手段 2 0 3 によって実現される。

## 【 0 0 8 4 】

ステップ 2 3 1 は、命令列記憶装置 2 0 1 から I 番目の命令を読み出す操作である。ステップ 2 3 2 とステップ 2 3 2 0 は、第 1 の命令検出手段 2 0 2 において実行され、プロセッサステータスを更新する可能性がある命令を検出し、それに相当する命令番号 I を配列 1 に検出された順番に従って格納する操作である。つまり、このステップ 2 3 2 とステップ 2 3 2 0 の処理により、記述プログラムの命令列からプロセッサの状態を更新する可能性がある命令を検出する第 1 の命令検出手段が実行される。

## 【 0 0 8 5 】

次に、ステップ 2 3 3 とステップ 2 3 3 0 は、第 2 の命令検出手段 2 0 3 において実行され、条件付きで実行される命令か検出し、それに相当する命令番号 I

を配列 2 に検出された順番に従って格納する操作である。つまり、このステップ 2 3 3 とステップ 2 3 3 0 の処理により、記述プログラムの命令列からプロセッサの状態に応じて実行するか否か決定される条件付きの命令を検出する第 2 の命令検出手段が実行される。

## 【 0 0 8 6 】

次に、ステップ 2 3 4 は、すべての命令が処理されたか判定する操作である。ステップ 2 3 5 は、次の命令番号 I を処理するために I を 1 増分する操作である。この第 1 の処理は、ステップ 2 3 0 で  $I = 1$  の命令番号から操作を開始し、すべての命令を処理した時点で完了する（ステップ 2 3 6 に相当）ものとする。

## 【 0 0 8 7 】

上記の第 1 の処理により、第 1 の命令検出手段および第 2 の命令検出手段を実行することができる。

## 【 0 0 8 8 】

図 1 0 は、本発明の第 2 実施形態におけるプログラム変換装置による第 2 の処理 2 0 7 を示す図である。この第 2 の処理 2 0 7 は、上記の第 1 の処理 2 0 6 において検出された条件・判断命令を、演算命令と該演算命令に後続し該演算命令に伴う一つまたは複数の後続命令との命令列に書き換える命令書き換え手段を実行する処理である。この第 2 の処理 2 0 7 は、第 1 の判定手段 2 0 8 と第 2 の判定手段 2 0 9 と命令検索手段 2 1 0 と命令書き換え手段 2 0 5 によって実現される。

## 【 0 0 8 9 】

ステップ 2 4 1 は、配列 1 から 1 つの命令番号 J を読み出す操作である。

## 【 0 0 9 0 】

ステップ 2 4 2 は、第 1 の判定手段 2 0 8 で実行され、配列 2 から  $J + 1$  と  $J + 2$  の命令番号を検索する操作である。つまり、ステップ 2 4 2 は第 1 の命令検出手段 2 0 2 により検出された命令に後続して、第 2 の命令検出手段 2 0 3 により検出された命令である第 1 の条件付き命令と第 2 の条件付き命令の 2 つの命令が連続しているかを判定する第 1 の判定手段を実行する処理である。

## 【 0 0 9 1 】

ステップ 2 4 3 は、命令列記憶装置から命令番号 J とステップ 2 4 2 で検索した命令を読み出し記憶する操作である（ここでは命令 A、命令 B、命令 C）。

【 0 0 9 2 】

ステップ 2 4 4 は、第 2 の判定手段 2 0 9 で実行され、前記命令 B と命令 C の条件が排他的な条件か判定する操作である（以下、条件 1 と条件 2 とする）。つまり、第 1 の条件付き命令と前記第 2 の条件付き命令の実行条件同士が互いに相反するものであるかを判定する第 2 の判定手段を実行する処理である。

【 0 0 9 3 】

ステップ 2 4 5 は、命令検索手段 2 1 0 で実行され、命令 A と同じ処理を実行可能で、かつ条件 1 と条件 2 を後続に配置可能で、かつ後続の 2 命令を対象とする指示命令が存在するか判定する操作である。つまり、このステップ 2 4 5 は、機械語命令コードの中から、第 1 の命令検出手段 2 0 2 により検出された命令と同一の処理実行を示し、かつ、後続する前記第 1 の条件付き命令および前記第 2 の条件付き命令が互いに相反する条件で実行されるものであることを示す指示命令を検索する指示命令検索手段を実行する処理である。

【 0 0 9 4 】

ステップ 2 4 6 は、命令書き換え手段 2 0 5 で実行され、ステップ 2 4 2 で読み出した命令を指示命令と無条件実行命令に置き換える操作である。

【 0 0 9 5 】

ステップ 2 4 7 は、ステップ 2 4 6 で置き換えた命令列で命令列記憶装置を書き換える操作である。つまり、このステップ 2 4 6 とステップ 2 4 7 は、第 1 の命令検出手段 2 0 2 と第 2 の命令検出手段 2 0 3 によって検出された命令列を、第 1 の条件付き命令から当該第 1 の条件記述を除いた第 1 の後続命令と、第 2 の条件付き命令から当該第 2 の条件記述を除いた第 2 の後続命令と、指示命令とに書き換える命令書き換え手段を実行する処理である。

【 0 0 9 6 】

ステップ 2 4 8 は、配列 1 に格納されたすべての命令番号 J を処理したか判定する操作である。ステップ 2 4 9 は、次の命令番号 J を対象とするために配列 1 の要素を 1 増分して、ステップ 2 4 1 に移る。ステップ 2 5 0 で処理を終了する

## 【0097】

以下、図7の書き換え前プログラムが本実施形態2のプログラム変換装置により書き換えられていく流れを第1の処理206および第2の処理207の順に説明する。なお、1行目の命令を命令(1)、2行目の命令を命令(2)というように括弧づけにより表現する。

## 【0098】

## (第1の処理206)

ステップ230では、 $I = 1$ として命令(1) (ADD R5,R1)が設定される。ステップ231では、命令(1) (ADD R5,R1)が命令列記憶装置201から読み出される。ステップ232では、第1の命令検出手段202において命令(1) (ADD R5,R1)がプロセッサステータスを更新する可能性がある命令であると検出され、ステップ2320に移り配列1の1番目に1が格納される。ステップ233では、第2の命令検出手段203において命令(1) (ADD R5,R1)が条件付きの命令であるとは検出されないので配列2は操作されない。

## 【0099】

ステップ234では、まだすべての命令を処理していないので、命令番号を1増やして(ステップ235)、ステップ231に戻り、命令(2) (SUB R0,R1)が命令列記憶装置201から読み出される。ステップ232では、第1の命令検出手段202において命令(2) (SUB R0,R1)がプロセッサステータスを更新する可能性がある命令であると検出され、ステップ2320に移り配列1の2番目に2が格納される。ステップ233では、第2の命令検出手段203において命令(2) (SUB R0,R1)が条件付き実行命令とは検出されないので配列2は操作されない。

## 【0100】

ステップ234では、まだすべての命令を処理していないので、命令番号を1増やして(ステップ235)、ステップ231に戻り、命令(3) (MOVIFZF R2,R4)が命令列記憶装置201から読み出される。ステップ232では、第1の命令検出手段202において命令(3) (MOVIFZF R2,R4)がプロセッサステータス

を更新する可能性がある命令とは検出されないので配列 1 は操作されない。ステップ 2 3 3 では、第 2 の命令検索手段 2 0 3 において命令 (3) (MOVIFZF R2,R4) が条件付きの命令であると検出され、ステップ 2 3 3 0 に移り配列 2 の 1 番目に 3 が格納される。

## 【 0 1 0 1 】

ステップ 2 3 4 では、まだすべての命令を処理していないので、命令番号を 1 増やして (ステップ 2 3 5)、ステップ 2 3 1 に戻り、命令 (4) (MOVIFNZF R3,R4) が命令列記憶装置 2 0 1 から読み出される。ステップ 2 3 2 では、第 1 の命令検索手段 2 0 2 において命令 (4) (MOVIFNZF R3,R4) がプロセッサステータスを更新する可能性がある命令でないので配列 1 は操作されない。ステップ 2 3 3 では、第 2 の命令検出手段 2 0 3 において命令 (4) (MOVIFNZF R3,R4) が条件付きの命令であると検出され、ステップ 2 3 3 0 に移り配列 2 の 2 番目に 4 が格納される。

## 【 0 1 0 2 】

以降、同様のステップを踏んで配列 1 の 3 番目には 5 が、配列 2 の 3 番目には 7 が格納される。その後、命令 (8) (MOV R6,R0) に対してステップ 2 3 2、ステップ 2 3 3 と進み、ステップ 2 3 4 ですべての命令を処理した後、第 1 の処理 2 0 6 が完了する。このとき、配列 1 には 1、2、5 が、配列 2 には 3、4、7 が格納されており、それらは第 2 の処理 2 0 7 で使用される。

## 【 0 1 0 3 】

(第 2 の処理 2 0 7)

ステップ 2 4 0 : 配列 1 の 1 番目の要素から処理を開始する。

## 【 0 1 0 4 】

ステップ 2 4 1 : 配列 1 から 1 番目の命令番号である 1 を読み出し変数 J に格納する。

## 【 0 1 0 5 】

ステップ 2 4 2 : 第 1 の判定手段 2 0 8 において、配列 2 から J + 1 と J + 2 の命令番号を検索する。いま J = 1 であるので命令 (2) と命令 (3) の命令番号の有無が調べられる。ここでは J + 1、つまり、命令 (2) の命令番号が存在

しないので第1の判定手段では条件成立が検出されず、ステップ248に移る。

【0106】

ステップ248：配列1のすべての命令番号を処理したか判定する。まだ残っているのでステップ249に移る。

【0107】

ステップ249：配列1における次の命令番号を処理するように、配列1の要素を1増分する。

【0108】

ステップ241：配列1から2番目の命令番号である2を読み出し変数Jに格納する。

【0109】

ステップ242：第1の判定手段208において、配列2から変数J+1と変数J+2の命令番号を検索する。いま、J=2であり、配列2に命令(3)と命令(4)の命令番号が存在するか調べることとなる。ここではどちらも存在するのでステップ243に移る。

【0110】

ステップ243：命令列記憶装置から変数J、変数J+1、変数J+2に対応する命令を読み出し、それぞれ命令A(SUB)、命令B(MOVIFZF)、命令C(MOVIFNZF)として記憶する。

【0111】

ステップ244：第2の判定手段209において、命令Bの条件1と命令Cの条件2が排他的な条件か判定する。ここではゼロフラグの状態が1のとき命令Bが実行され、ゼロフラグの状態が0のとき命令Cが実行されることを検出し、命令Bと命令Cがゼロフラグの状態において相反するとき実行されるので排他的な条件と判定する。また、条件1を「排他的条件が1である」として、条件2を「排他的条件が0である」として記憶する。次に、ステップ245に移る。

【0112】

ステップ245：命令検索手段210において、命令Aと同じ処理を実行可能で、かつ条件1と条件2を後続に配置可能で、かつ後続の2命令を対象とする指



示命令が存在するか判定する。ここではSUBIFEZFが候補として存在するので、ステップ246に移る。

【0113】

ステップ246：命令書き換え手段205において、命令AをSUBIFEZFに置き換える。また命令Bおよび命令Cに対応する処理を無条件に実行する命令に置き換える。ここでは命令BをMOVに、命令CをMOVに置き換える。次にステップ247に移る。

【0114】

ステップ247：命令書き換え手段205において、命令列記憶装置の変数J、変数J+1、変数J+2に対応する命令を、それぞれ命令A（SUBIFEZF）、命令B（MOV）、命令C（MOV）に書き換える。

【0115】

ステップ248：配列1のすべての命令番号を処理したか判定する。まだ残っているのでステップ249に移る。

【0116】

ステップ249：配列1における次の命令番号を処理するように、配列1の要素を1増分する。

【0117】

ステップ241：配列1から3番目の命令番号である5を読み出し変数Jに格納する。

【0118】

ステップ242：第1の命令検出手段202において、配列2から変数J+1と変数J+2の命令番号を検索する。いまJ=5であり、命令（6）と命令（7）の命令番号が存在するか否かを調べる。ここでは7の命令番号は存在するが6の命令番号が存在しないので、次にステップ248に移る。

【0119】

ステップ248：配列1のすべての命令番号を処理したか判定する。配列1にもう命令番号は存在しないのでステップ250に移る。

【0120】

ステップ 2 5 0 : 第 2 の処理 2 0 7 を完了する。

【 0 1 2 1 】

以上のように、本実施形態 2 によれば、命令列記憶装置 2 0 1 に格納された書き換え前のプログラムである図 7 の命令 ( 2 ) と命令 ( 3 ) と命令 ( 4 ) が、図 8 の命令 ( 2 ) と命令 ( 3 ) と命令 ( 4 ) のように書き換えられる。

【 0 1 2 2 】

なお、本実施形態 2 によるプログラム変換装置は、ステップ 2 4 4 で 2 つの条件が排他的か否か判定し、ステップ 2 4 5 で 2 命令の 1 つ前の命令を、後続の 2 命令を排他的な条件付き実行を指示する指示命令に置き換えることが可能か判定している。このように、プログラム変換装置 6 2 は、命令列記憶装置 2 0 1 に最終的に格納される命令列を、後続の命令 1 とその後に配置された命令 2 を互いに相反する条件が成立するとき実行する命令として指示する指示命令を含む命令列によって書き換えることが可能である。

【 0 1 2 3 】

なお、本実施形態 2 のプログラム変換装置にはアセンブリ言語プログラムを入力するものとしたが、機械語プログラムを入力として新たな機械語プログラムを生成するプログラム変換装置としてもよい。

【 0 1 2 4 】

( 実施形態 3 )

本実施形態 3 のデータ処理装置は、本発明の第 1 および第 2 のデータ処理装置の例である。実施形態 1 では後続命令が 1 つの命令であったが、本実施形態 3 は、後続命令が複数の命令列からなる後続命令群である。

【 0 1 2 5 】

以下、本発明の実施形態 3 について図 1、図 2、図 1 1、図 1 2、図 1 3 を用いて説明する。本実施形態 3 は、条件付き実行ステータス 7 0 9 の更新を操作することで、実行するための条件が明示されない複数の命令列を条件付き実行するものである。なお、実施の形態 1 と同様の部分については、同一図と符号を付与して説明を省略する。

【 0 1 2 6 】

本実施形態 3 のデータ処理装置の全体構成例に関しては図 1、プロセッサの構成例に関しては図 2 で良く、実施形態 1 と同様であるのでここでの説明は省略する。

## 【 0 1 2 7 】

本実施形態における命令無効化制御方法と条件付き実行ステータス 7 0 9 の遷移について図 1 1 を用いて説明する。図 1 1 ( a ) は、命令無効化制御回路 7 0 8 が排他的条件と条件付き実行ステータスに応じた EX ステージの無効化処理を表している ( 命令無効化信号 7 0 7 ) 。図 1 1 の排他的条件 7 0 1 とは、指示命令によって選択されたプロセッサステータスの状態であり “ 0 ” か “ 1 ” の 2 値をとる。図 1 1 の条件付き実行ステータス 7 0 9 とは、同じく指示命令によって後続の命令を条件付き実行している状態を表している。図 1 1 ( a ) に従えば、例えば、排他的条件 7 0 1 が “ 1 ” で条件付き実行ステータス 7 0 9 が “ x' 100 ” のときは EX ステージの処理が無効化される。図 1 1 ( b ) は、条件付き実行ステータス 7 0 9 の遷移図であり、シーケンサ 7 1 6 により制御される。条件付き実行ステータス 7 0 9 は、初期値 “ x' 000 ” から前記の指示命令が解読された後 “ x' 101 ” にセットされ、次命令が解読されると “ x' 1 ” ずつデクリメントしてゆき、“ x' 000 ” まで順に更新される。

## 【 0 1 2 8 】

以上が本実施形態 3 のデータ処理装置の構成例である。以下に本実施形態 3 のデータ処理装置の動作の例を説明する。一例として図 1 2 に示す機械語プログラム 2 を実行処理する場合の動作を説明する。図 1 2 は機械語プログラム 2 を表しており、ニーモニック表記である。その命令について簡単な説明を付する。

## 【 0 1 2 9 】

( 機械語プログラム 2 )

1 行目 : SUBIFEXZF R0,R1 ( R0 レジスタの値から R1 のレジスタの値を減算して結果を R1 レジスタに格納する。さらに後続の 4 命令をプロセッサステータスのゼロフラグを条件として排他的な条件付き実行命令として指示する )

2 行目 : MOV R2,R6 ( R2 レジスタの値を R6 レジスタに格納する )

3 行目 : MOV R3,R7 ( R3 レジスタの値を R7 レジスタに格納する )

4 行目 : MOV R4,R6 (R4レジスタの値をR6レジスタに格納する)

5 行目 : MOV R5,R7 (R5レジスタの値をR7レジスタに格納する)

6 行目 : MOV x' 0001,R0 (即値x' 0001をR6レジスタに格納する)

7 行目 : ADD R0,R6 (R0レジスタの値とR6レジスタの値を加算して結果をR6レジスタに格納する)

8 行目 : NOP (何も実行しない)

このように、上記の機械語プログラム2において、1行目の指示命令SUBIFEXZFは、後続する2行目から5行目に配置された命令群の一方のみの命令群を実行するという排他的な条件付き命令となっている。指示命令SUBIFEXZFの演算実行結果に基づいてゼロフラグの状態が決定され、当該ゼロフラグが示す排他的条件701の状態と条件付き実行ステータス709に応じて、後続する2行目、3行目の命令群と4行目、5行目の命令群の一方の命令群のみが排他的に実行され、他方は無効化されることとなる。

#### 【0130】

図13はプロセッサ64の動作タイミング図を示すものである。また、図13に示すようにステージの前半（例えばタイミングt11）と後半（タイミングt12）を表すタイミングにより命令無効化制御回路708は動作するものとする。特にDECステージではタイミングの後半で命令無効化制御回路708により条件付き実行ステータス709を更新するものとする。ここで、時間が経過する順にタイミング毎に図12の機械語プログラムを実行する場合の動作を説明する。以下、時間が経過する順にタイミング毎にその動作を説明する。なお、R0とR1には同値である“x' 0001”が格納され、R2には“x' 0002”、R3には“x' 0003”、R4には“x' 0004”、R5には“x' 0005”が格納されていたとする。

#### 【0131】

(タイミングt1)

IFステージ : 命令1 [SUBIFEXZF R0,R1]

タイミングt11 : 命令1がROMから読み出され、命令ラッチ714に格納される。

#### 【0132】

(タイミング t 2)

DECステージ：命令 1 [SUBIFEXZF R0,R1]

タイミング t 2 1：命令ラッチ 7 1 4 に格納された命令 1 が命令解読器 7 0 5 で解読され、R0レジスタから値が読出され D 1 ラッチ 7 1 0 に “x’ 0001” が格納される。R1レジスタから値が読み出され D 2 ラッチ 7 1 1 に “x’ 0001” が格納される。

【 0 1 3 3 】

タイミング t 2 2：命令無効化制御回路 7 0 8 は、条件付き実行ステータス 7 0 9 を “x’ 000” から “x’ 101” にセットする。排他的条件 7 0 1 の値を表わすレジスタとしてゼロフラグがセクタ 7 0 2 により選択される。

【 0 1 3 4 】

IFステージ：命令 2 [MOV R2,R6]

タイミング t 2 1：命令 2 が ROM 7 1 7 から読出され、命令ラッチ 7 1 4 に格納される。

【 0 1 3 5 】

(タイミング t 3)

EXステージ：命令 1 [SUBIFEXZF R0,R1]

タイミング t 3 1：命令無効化信号 7 0 7 は初期状態のまま “0” となっており、EXステージの実行が行われ、命令 1 の実行が開始される。D 1 ラッチ 7 1 0 と D 2 ラッチ 7 1 1 に格納された値が演算器 7 0 3 で減算される。ここでは D 1 ラッチ 7 1 0 と D 2 ラッチ 7 1 1 には同値 “x’ 0001” が格納されているので、減算結果として R1レジスタに “x’ 0000” が格納される。この結果ゼロフラグには “1” が格納される。

【 0 1 3 6 】

DECステージ：命令 2 [MOV R2,R6]

タイミング t 3 1：命令ラッチ 7 1 4 に格納された命令 2 が命令解読器 7 0 5 で解読され、R2レジスタから値が読み出され D 1 ラッチ 7 1 0 に “x’ 0002” が格納される。

【 0 1 3 7 】

タイミング t 3 2 : 命令無効化制御回路 7 0 8 は、条件付き実行ステータス 7 0 9 が “x'101” から “x'100” に更新する。いま、排他的条件は “1” であり、条件付き実行ステータスが “x'100” であるので、図 1 1 (a) に従い、命令無効化信号 7 0 7 は “0” のままとなる。

【0 1 3 8】

IFステージ：命令 3 [MOV R3,R7]

タイミング t 3 1 : 命令 3 が ROM 7 1 7 から読出され、命令ラッチ 7 1 4 に格納される。

【0 1 3 9】

(タイミング t 4)

EXステージ：命令 2 [MOV R2,R6]

タイミング t 4 1 : 命令無効化信号 7 0 7 は “0” であるので、命令 2 の実行が開始される。D 1 ラッチ 7 1 0 に格納された値 “x'0002” が読み出され演算器 7 0 3 をスルーして R6 レジスタに格納される。つまり、この例では後続命令のうち命令 2 が実行された。

【0 1 4 0】

DECステージ：命令 3 [MOV R3,R7]

タイミング t 4 1 : 命令ラッチ 7 1 4 に格納された命令 3 が命令解読器 7 0 5 で解読され、R3 レジスタから値が読出され D 1 ラッチ 7 1 0 に “x'0003” が格納される。R7 レジスタから値が読み出され D 2 ラッチ 7 1 1 に格納される。

【0 1 4 1】

タイミング t 4 2 : 命令無効化制御回路 7 0 8 は、条件付き実行ステータス 7 0 9 を “x'100” から “x'011” に更新する。いま、排他的条件は “1” であり、条件付き実行ステータスが “x'011” であるので、図 1 1 (a) に従い、命令無効化信号 7 0 7 は “0” のままとなる。

【0 1 4 2】

IFステージ：命令 4 [MOV R4,R6]

タイミング t 4 1 : 命令 4 が ROM 7 1 7 から読出され、命令ラッチ 7 1 4 に格納される。

## 【 0 1 4 3 】

( タイミング t 5 )

EXステージ：命令 3 [MOV R3,R7]

タイミング t 5 1：命令無効化信号 7 0 7 は “ 0 ” であるので命令 3 の実行が開始される。D 1 ラッチ 7 1 0 に格納された値 “x' 0003” が読み出され演算器 7 0 3 をスルーして R7 レジスタに格納される。つまり、この例では後続する命令のうち命令 3 は実行される。

## 【 0 1 4 4 】

DECステージ：命令 4 [MOV R4,R6]

タイミング t 5 1：命令ラッチ 7 1 4 に格納された命令 4 が命令解読器 7 0 5 で解読され、R4 レジスタから値 “x' 0004” が読み出され D 1 ラッチ 7 1 0 に格納される。

## 【 0 1 4 5 】

タイミング t 5 2：命令無効化制御回路 7 0 8 は、条件付き実行ステータス 7 0 9 を “x' 011” から “x' 010” に更新する。ここで、命令無効化制御回路 7 0 8 は、図 1 1 ( a ) に従って、条件付き実行ステータスが “x' 010” であり、排他的条件が “1” であるので命令無効化信号 7 0 7 を生成し、命令無効化信号 7 0 7 を “0” から “1” に更新する。

## 【 0 1 4 6 】

IFステージ：命令 5 [MOV R5,R7]

タイミング t 5 1：命令 5 が ROM 7 1 7 から読出され、命令ラッチ 7 1 4 に格納される。

## 【 0 1 4 7 】

( タイミング t 6 )

EXステージ：命令 4 [MOV R4,R6]

タイミング t 6 1：命令無効化信号 7 0 7 が “1” であるので命令 4 は無効化される。つまり、この例では後続する命令のうち、命令 2 と命令 3 は実行されたが、命令 4 は無効化されて実行されない。

## 【 0 1 4 8 】

DECステージ：命令 5 [MOV R5,R7]

タイミング t 6 1：命令ラッチ 7 1 4 に格納された命令 5 が命令解読器 7 0 5 で解読され、R5レジスタから値 “x' 0005” が読み出され D 1 ラッチ 7 1 0 に格納される。

【 0 1 4 9 】

タイミング t 6 2：命令無効化制御回路 7 0 8 は、条件付き実行ステータス 7 0 9 を “x' 010” から “x' 001” に更新する。ここで、命令無効化制御回路 7 0 8 は、図 1 1 (a) に従って、条件付き実行ステータスが “x' 001” であり、排他的条件が “1” であるので命令無効化信号 7 0 7 を生成し、命令無効化信号 7 0 7 “1” のままとなる。

【 0 1 5 0 】

IFステージ：命令 6 [MOV x' 0001,R6]

タイミング t 6 1：命令 6 が R O M から読出され、命令ラッチ 7 1 4 に格納される。

【 0 1 5 1 】

(タイミング t 7)

EXステージ：命令 5 [MOV R5,R7]

タイミング t 7 1：命令無効化信号 7 0 7 が “1” であるので命令 5 は無効化される。つまり、この例では後続する命令のうち、命令 2 と命令 3 は実行されたが、命令 4 と命令 5 は無効化されて実行されない。

【 0 1 5 2 】

DECステージ：命令 6 [MOV x' 0001,R6]

タイミング t 7 1：命令ラッチ 7 1 4 に格納された命令 6 が命令解読器 7 0 5 で解読され、D 1 ラッチ 7 1 0 に即値である x' 0001 が格納される。

【 0 1 5 3 】

タイミング t 7 2：命令無効化制御回路 7 0 8 は、条件付き実行ステータス 7 0 9 を “x' 001” から “x' 000” に更新する。ここで、命令無効化制御回路 7 0 8 は図 1 1 (a) に従って、条件付き実行ステータスが “x' 000” であり、排他的条件が “1” であるので、命令無効化信号 7 0 7 を “1” から “0” に更新する



【0154】

IFステージ：命令7 [ADD R0,R6]

タイミングt 71：命令7がROM 717から読出され、命令ラッチ714に格納される。

【0155】

(タイミングt 8)

EXステージ：命令6 [MOV x'0001,R6]

タイミングt 81：命令無効化信号707は“0”であるので命令6の実行を開始する。D1ラッチ710に格納された値が読み出され、演算器703をスルーしてR6レジスタに格納される。

【0156】

DECステージ：命令7 [ADD R0,R6]

タイミングt 81：命令ラッチ714に格納された命令7が命令解読器705で解読される。R0レジスタから値“x'0001”が読み出されD1ラッチ710に格納される。R6レジスタから値“x'0001”が読み出されD2ラッチ711に格納される。

【0157】

IFステージ：命令8 [NOP]

タイミングt 81：命令8がROM 717から読出され、命令ラッチ714に格納される。

【0158】

以上のように本実施の形態によれば、命令1 [SUBIFEXZF R0,R1]により排他的な条件としてゼロフラグを選択し、条件付き実行ステータス709をセットすることで条件付き実行されるので、複数の命令からなる命令群1（命令2と命令3）と命令群2（命令4と命令5）には特別な条件指定フィールドを付加する必要が無く、いずれか一方の命令群のみが実行され、他方は無効化されて実行されない。

【0159】

なお、上記の説明ではR0レジスタとR1レジスタの値を同値として、ゼロフラグの値が“1”となる場合を示したが、R0レジスタとR1レジスタの値が同値ではなく“0”となる場合も、命令無効化制御回路708は図11(a)に従って命令無効化信号707を制御することにより命令無効化制御を実行することができる。この場合は後続命令のうち命令2と命令3が無効化されて実行されず、命令4と命令5が実行される。

## 【0160】

また、上記に示した例では、命令1はプロセッサステータスワード712のうちゼロフラグを命令無効化信号707として選択する命令としたが、限定されるものではなく、他のプロセッサステータスワード712を選択しても良い。

## 【0161】

## (実施形態4)

実施形態4として、本発明の第1および第2のプログラム変換装置の例を示す。実施形態2では後続命令が1つの命令であったが、本実施形態4は、後続命令が複数の命令列からなる後続命令群である。

## 【0162】

本発明のプログラム変換装置を図6、図12、図14、図15、図16によって説明する。既に説明した図には同一番号を付して説明を省略する。

## 【0163】

本実施形態4のデータ処理装置の全体構成例に関しては図6で良く、実施形態2と同様であるのでここでの説明は省略する。

## 【0164】

本実施形態4のプログラム変換装置によるプログラムの書き換え動作の例を説明する。以下に2つの書き換え例を示す。1つ目は、図14の書き換え前のアセンブリ言語プログラムを図12の機械語プログラムに書き換える例を示す。図12の機械語プログラムは、実施形態3で用いたプログラムと同様である。2つ目は、図15の書き換え前のアセンブリ言語プログラムを図12に書き換える例である。図12、図14、図15のニーモニック表記した命令はすでに説明されており、ADD命令、MOV命令などで使用しているオペランド（即値、レジスタ）が異

なるだけであるのでここでは説明を省略する。

【0165】

図16は、本実施形態4におけるプログラム変換装置による第2の処理207の前半部を示す図である。この第2の処理207は、第1の処理206において検出された条件・判断命令を、演算命令と該演算命令に後続し該演算命令に伴う一つまたは複数の後続命令群との命令列に書き換える命令書き換え手段を実行する処理である。この第2の処理207は、第1の判定手段208と第2の判定手段209と命令検索手段210と命令書き換え手段205によって実現される。なお、同図の211aは図10の211に該当する部分を示しており、それ以外は図10と同様であるとしてここでの説明を省略する。

【0166】

ステップ242aは、第1の判定手段208で実行され、配列2からJ+1、J+2、J+3、J+4の命令番号を検索する操作である。つまり、ステップ242aは第1の命令検出手段202により検出された命令に後続して、第2の命令検出手段203により検出された命令である第1の条件付き命令と第2の条件付き命令の2つの命令が連続しているかを判定する第1の判定手段を実行する処理である。

【0167】

ステップ243aは、命令番号に相当する命令を読み出し記憶する操作である（以下、命令A、命令B、命令C、命令D、命令Eとする）。

【0168】

ステップ244aは、第2の判定手段209で実行され、命令Aと命令Bの条件1と命令Cと命令Dの条件2が排他的な条件か判定する操作である。つまり、第1の条件付き命令と前記第2の条件付き命令の実行条件同士が互いに相反するものであるかを判定する第2の判定手段を実行する処理である。

【0169】

ステップ245aは、命令検索手段210において、命令Aと同じ処理を実行可能で、かつ条件1と条件2を後続に配置可能で、かつ後続の4命令を対象とする指示命令が存在するか判定する操作である。つまり、このステップ245aは

、機械語命令コードの中から、第 1 の命令検出手段 2 0 2 により検出された命令と同一の処理実行を示し、かつ、後続する前記第 1 の条件付き命令および前記第 2 の条件付き命令が互いに相反する条件で実行されるものであることを示す指示命令を検索する指示命令検索手段を実行する処理である。

【0 1 7 0】

以上が本実施形態 4 の 2 1 1 a の処理である。

【0 1 7 1】

まず、始めに、図 1 4 の書き換え前アセンブリ言語プログラムを図 1 2 の機械語プログラムに書き換える動作を第 1 の処理 2 0 6 と第 2 の処理 2 0 7 の順番に説明する。なお、1 行目の命令を命令 (1)、2 行目の命令を命令 (2) のように括弧で以下表現する。

【0 1 7 2】

(第 1 の処理 2 0 6)

実施形態 2 と同様に処理され、最終的に、配列 1 には 1、7 が、配列 2 には 2、3、4、5 が格納されており、それらは第 2 の処理 2 0 7 で使用される。

【0 1 7 3】

(第 2 の処理 2 0 7)

ステップ 2 4 0 : 配列 1 の 1 番目の要素から処理を開始する。

【0 1 7 4】

ステップ 2 4 1 : 配列 1 から 1 番目の命令番号である 1 を読み出し変数 J に格納する。

【0 1 7 5】

ステップ 2 4 2 a : 第 1 の判定手段 2 0 8 において、配列 2 から J + 1 と J + 2 と J + 3 と J + 4 の命令番号を検索する。いま J = 1 であるので、命令 (2)、命令 (3)、命令 (4)、命令 (5) のすべてが存在するのでステップ 2 4 3 に移る。

【0 1 7 6】

ステップ 2 4 3 a : 命令列記憶装置から変数 J から J + 4 に対応する命令を読み出し、それぞれ命令 A (SUB)、命令 B (MOVIFZF)、命令 C (MOVIFZF)、命令 D (MO

VIFNZF)、命令 E (MOVIFNZF)として記憶する。

【0177】

ステップ 244 a : 第 2 の判定手段 209 において、4 命令のうち、2 命令の条件が条件 1 であり、他の 2 命令の条件が条件 2 であり、条件 1 と条件 2 が排他的な条件か判定する。ここではゼロフラグの状態が 1 のとき命令 B と命令 C が実行され、ゼロフラグの状態が 0 のとき命令 D と命令 E が実行されることを検出し、ゼロフラグの状態において相反するとき実行されるので排他的な条件と判定する。また、条件 1 を「排他的条件が 1 である」として、条件 2 を「排他的条件が 0 である」として記憶する。

【0178】

ステップ 245 a : 命令検索手段 210 において、命令 A と同じ処理を実行可能で、かつ条件 1 と条件 2 を後続に配置可能で、かつ後続の 4 命令を対象とする指示命令が存在するか判定する。ここでは SUBEXIFZF が候補として存在するので、ステップ 246 に移る。

【0179】

ステップ 246 : 命令書き換え手段 205 において、命令 A を SUBEXIFZF に置き換える。また命令 B、命令 C、命令 D、命令 E に対応する処理を無条件に実行する命令に置き換える。次にステップ 247 に移る。

【0180】

ステップ 247 : 命令書き換え手段 205 において、命令列記憶装置の変数 J から J + 4 に対応する命令を、それぞれ命令 A (SUBIFEZF)、命令 B (MOV)、命令 C (MOV)、命令 D (MOV)、命令 E (MOV) に書き換える。

【0181】

ステップ 248 : 配列 1 のすべての命令番号を処理したか判定する。まだ残っているのでステップ 249 に移る。

【0182】

ステップ 249 : 配列 1 における次の命令番号を処理するように、配列 1 の要素を 1 増分する。

【0183】

ステップ 2 4 1 : 配列 1 から 2 番目の命令番号である 7 を読み出し変数 J に格納する。

【 0 1 8 4 】

ステップ 2 4 2 : 第 1 の命令検出手段 2 0 2 において、配列 2 から変数 J + 1 から J + 4 の命令番号を検索する。いま J = 7 であり、命令 ( 8 ) から命令 ( 1 1 ) の命令番号が存在するか否かを調べる。ここでは存在しないので、次にステップ 2 4 8 に移る。

【 0 1 8 5 】

ステップ 2 4 8 : 配列 1 のすべての命令番号を処理したか判定する。配列 1 にもう命令番号は存在しないのでステップ 2 5 0 に移る。

【 0 1 8 6 】

ステップ 2 5 0 : 第 2 の処理 2 0 7 を完了する。

【 0 1 8 7 】

以上のように、本実施形態 4 によれば、命令列記憶装置 2 0 1 に格納された書き換え前のプログラムである図 1 4 の命令 ( 1 ) から命令 ( 5 ) が、図 1 2 の命令 ( 1 ) から命令 ( 5 ) に書き換えられる。なお、本実施形態 4 によるプログラム変換装置は、ステップ 2 4 4 で 2 つの条件が排他的か否か判定し、ステップ 2 4 5 で 4 命令の 1 つ前の命令を、後続の 4 命令を排他的な条件付き実行を指示する指示命令に置き換えることが可能か判定している。このように、プログラム変換装置 6 2 は、命令列記憶装置 2 0 1 に最終的に格納される命令列を、後続の命令列 1 とその後に配置された命令列 2 を互いに相反する条件が成立するとき実行する命令として指示する指示命令を含む命令列に書き換えることが可能である。

【 0 1 8 8 】

なお、本実施形態 4 のプログラム変換装置にはアセンブリ言語プログラムを入力するものとしたが、機械語プログラムを入力として新たな機械語プログラムを生成するプログラム変換装置としてもよい。

【 0 1 8 9 】

次に、図 1 5 の書き換え前アセンブリ言語プログラムを図 1 2 の機械語プログラムに書き換える動作を説明する。図 1 5 では、条件付きの実行命令が 2 行目、

3行目、4行目、5行目は条件付きの実行命令であるが、2行目と5行目の条件1と、3行目と4行目の条件2が、互いに排他的な条件となっている。

【0190】

以下に、第1の処理206と第2の処理207の順番に説明する。なお、1行目の命令を命令(1)、2行目の命令を命令(2)のように括弧で以下表現する。

【0191】

(第1の処理206)

実施形態2と同様に処理され、最終的に、配列1には1、7が、配列2には2、3、4、5が格納されており、それらは第2の処理207で使用される。

【0192】

(第2の処理207)

ステップ240：配列1の1番目の要素から処理を開始する。

【0193】

ステップ241：配列1から1番目の命令番号である1を読み出し変数Jに格納する。

【0194】

ステップ242a：第1の判定手段208において、配列2からJ+1とJ+2とJ+3とJ+4の命令番号を検索する。いまJ=1であるので、命令(2)、命令(3)、命令(4)、命令(5)のすべてが存在するのでステップ243に移る。

【0195】

ステップ243a：命令列記憶装置から変数JからJ+4に対応する命令を読み出し、それぞれ命令A(SUB)、命令B(MOVIFNZF)、命令C(MOVIFZF)、命令D(MOVIFZF)、命令E(MOVIFNZF)として記憶する。

【0196】

ステップ244a：第2の判定手段209において、4命令のうち、2命令の条件が条件1であり、他の2命令の条件が条件2であり、条件1と条件2が排他的な条件か判定する。ここではゼロフラグの状態が1のとき命令Cと命令Dが実

行され、ゼロフラグの状態が0のとき命令Bと命令Eが実行されることを検出し、ゼロフラグの状態において相反するとき実行されるので排他的な条件と判定する。また、条件1を「排他的条件が1である」として、条件2を「排他的条件が0である」として記憶する。

## 【0197】

ステップ245a：命令検索手段210において、命令Aと同じ処理を実行可能で、かつ条件1と条件2を後続に配置可能で、かつ後続の4命令を対象とする指示命令が存在するか判定する。ここではSUBEXIFZFが候補として存在するので、ステップ246に移る。

## 【0198】

ステップ246：命令書き換え手段205において、命令AをSUBEXIFZFに置き換える。また命令B、命令C、命令D、命令Eに対応する処理を無条件に実行する命令に置き換える。さらに条件1に対応する命令を先に実行するよう並べ替える。ここでは、命令Cが命令Bに、命令Dが命令Cに、命令Bが命令Dに並べ替えられる。

## 【0199】

ステップ247：命令書き換え手段205において、命令列記憶装置の変数JからJ+4に対応する命令を、それぞれ命令A (SUBIFEZF)、命令B (MOV)、命令C (MOV)、命令D (MOV)、命令E (MOV)に書き換える。

## 【0200】

ステップ248：配列1のすべての命令番号を処理したか判定する。まだ残っているのでステップ249に移る。

## 【0201】

ステップ249：配列1における次の命令番号を処理するように、配列1の要素を1増分する。

## 【0202】

ステップ241：配列1から2番目の命令番号である7を読み出し変数Jに格納する。

## 【0203】



ステップ 2 4 2 : 第 1 の命令検出手段 2 0 2 において、配列 2 から変数  $J + 1$  から  $J + 4$  の命令番号を検索する。いま  $J = 7$  であり、命令 (8) から命令 (11) の命令番号が存在するか否かを調べる。ここでは存在しないので、次にステップ 2 4 8 に移る。

【 0 2 0 4 】

ステップ 2 4 8 : 配列 1 のすべての命令番号を処理したか判定する。配列 1 にもう命令番号は存在しないのでステップ 2 5 0 に移る。

【 0 2 0 5 】

ステップ 2 5 0 : 第 2 の処理 2 0 7 を完了する。

【 0 2 0 6 】

以上のように、本実施の形態によれば、命令列記憶装置 2 0 1 に格納された図 1 5 の命令 (1) から命令 (5) が、図 1 2 の命令 (1) から命令 (5) に書き換えられる。ここで、先に示したステップ 2 4 6 では、まず条件 1 に対応する命令 (命令 C、命令 D) を、その相対的な位置を変えずに命令番号  $J$  の直後に並べ替えている。次に条件 2 に対応する命令 (命令 B、命令 E) を、その相対的な位置を変えずに後方に並べ替えている。このとき条件 1 に対応する命令の実行順序は変更されていない。また、条件 2 に対応する命令の実行順序も変更されていない。このように、プログラム変換装置 6 2 は、指示命令により実行する条件が明示される命令列を、実現される処理には影響を及ぼさずに並べ替えることが可能であり、最終的に命令列記憶装置 2 0 1 に格納される命令列は、後続の命令列 1 とその後に配置された命令列 2 を互いに相反する条件が成立するとき実行する命令として指示する指示命令を含む命令列に書き換えられる。

【 0 2 0 7 】

(実施形態 5)

実施形態 5 として、本発明の第 3 および第 4 のデータ処理装置の例を示す。本実施形態 5 は、条件付き実行ステータス 7 0 9 の更新を操作することで、実行するための条件が明示されない命令を条件付き実行するものである。

【 0 2 0 8 】

実施形態 5 のデータ処理装置について図 1、図 2、図 1 7、図 1 8、図 1 9 を

用いて説明する。なお、実施形態1などと同様の部分については、同一図と符号を付与して説明を省略する。

#### 【0209】

本実施形態5のデータ処理装置の全体構成例に関しては図1、プロセッサの構成例に関しては図2で良く、実施形態1と同様であるのでここでの説明は省略する。

#### 【0210】

本実施形態5における命令無効化制御方法と条件付き実行ステータス709の遷移について図17を用いて説明する。図17(a)は、命令無効化制御回路708が排他的条件と条件付き実行ステータスに応じたEXステージの無効化処理を表している(命令無効化信号707)。同図の排他的条件701とは、指示命令によって選択されたプロセッサステータスの状態であり“0”か“1”の2値をとる。同図の条件付き実行ステータス709とは、同じく指示命令によって後続の命令を条件付き実行している状態を表している。例えば、図17(a)に従えば、排他的条件701が“1”で条件付き実行ステータス709が“x'01”のときはEXステージの処理が無効化される。図17(b)は、条件付き実行ステータス709の遷移図であり、シーケンサ716により制御される。条件付き実行ステータス709は、初期値“x'00”から前記の指示命令が解読された後“x'10”にセットされ、次命令が解読されると“x'1”ずつデクリメントしてゆき、“x'00”まで順に更新される。

#### 【0211】

以上が本実施形態5のデータ処理装置の構成例である。以下に本実施形態5のデータ処理装置の動作の例を説明する。一例として図18に示す機械語プログラム3を実行処理する場合の動作を説明する。図18は機械語プログラム3を表しており、ニーモニック表記である。その命令について簡単な説明を付する。

#### 【0212】

(機械語プログラム3)

1行目: SUBIFXZF R0,R1 (R0レジスタの値からR1のレジスタの値を減算して結果をR1レジスタに格納する。さらに後続の命令をプロセッサステータスのゼロフ

ラグが 0 のとき実行する条件付き実行命令として指示する)

2 行目 : MOV R2,R3 (R2レジスタの値をR3レジスタに格納する)

3 行目 : ADD R3,R6 (R3レジスタの値とR6レジスタの値を加算して結果をR6レジスタに格納する)

4 行目 : MOV R4,R7 (R4レジスタの値をR7レジスタに格納する)

このように、上記機械語プログラム 3 において、1 行目の指示命令 SUBIFXZF は後続する 2 行目に配置された命令がゼロフラグの状態に応じて実行される条件付き命令であることを指示するものとなっている。後述するように指示命令 SUBIFEXZF の演算実行結果に基づいてゼロフラグの状態が決定され、当該ゼロフラグが示す排他的条件 7 0 1 の状態と条件付き実行ステータス 7 0 9 に応じて、後続する 2 行目の命令が実行されるか無効化されるかが決められることとなる。

#### 【 0 2 1 3 】

図 1 9 はプロセッサ 6 4 の動作タイミング図を示すものである。また、図 1 9 に示すようにステージの前半（例えばタイミング t 1 1）と後半（タイミング t 1 2）を表すタイミングにより命令無効化制御回路 7 0 8 は動作するものとする。特に DEC ステージではタイミングの後半で命令無効化制御回路 7 0 8 により条件付き実行ステータス 7 0 9 を更新するものとする。ここで、時間が経過する順にタイミング毎に図 1 9 を実行する場合の動作を説明する。以下、時間が経過する順にタイミング毎にその動作を説明する。なお、R0 と R1 には同値である “x’ 00 01” が格納され、R2 には “x’ 0002”、R3 には “x’ 0003”、R4 には “x’ 0004” が格納されていたとする。

#### 【 0 2 1 4 】

(タイミング t 1)

IF ステージ : 命令 1 [SUBIFXZF R0,R1]

タイミング t 1 1 : 命令 1 が ROM 7 1 7 から読出され、命令ラッチ 7 1 4 に格納される。

#### 【 0 2 1 5 】

(タイミング t 2)

DEC ステージ : 命令 1 [SUBIFXZF R0,R1]

タイミング t 2 1 : 命令ラッチ 7 1 4 に格納された命令 1 が命令解読器 7 0 5 で解読され、R0レジスタから値が読出され D 1 ラッチ 7 1 0 に “x’ 0001” が格納される。R1レジスタから値が読み出され D 2 ラッチ 7 1 1 に “x’ 0001” が格納される。

## 【 0 2 1 6 】

タイミング t 2 2 : 命令無効化制御回路 7 0 8 は、条件付き実行ステータス 7 0 9 を “x’ 00” から “x’ 10” にセットする。排他的条件 7 0 1 の値を表わすレジスタとしてゼロフラグがセレクタ 7 0 2 により選択される。

## 【 0 2 1 7 】

IFステージ：命令 2 [MOV R2,R3]

タイミング t 2 1 : 命令 2 が ROM 7 1 7 から読出され、命令ラッチ 7 1 4 に格納される。

## 【 0 2 1 8 】

(タイミング t 3 )

EXステージ：命令 1 [SUBIFXZF R0,R1]

タイミング t 3 1 : 命令無効化信号 7 0 7 は初期状態のまま “0” となっており、EXステージの実行が行われ、命令 1 の実行が開始される。D 1 ラッチ 7 1 0 と D 2 ラッチ 7 1 1 に格納された値が演算器 7 0 3 で減算される。ここでは D 1 ラッチ 7 1 0 と D 2 ラッチ 7 1 1 には同値 “x’ 0001” が格納されているので、減算結果として R1レジスタに “x’ 0000” が格納される。この結果、ゼロフラグには “1” が格納される。

## 【 0 2 1 9 】

DECステージ：命令 2 [MOV R2,R3]

タイミング t 3 1 : 命令ラッチ 7 1 4 に格納された命令 2 が命令解読器 7 0 5 で解読され、R2レジスタから値が読出され D 1 ラッチ 7 1 0 “x’ 0002” が格納される。

## 【 0 2 2 0 】

タイミング t 3 2 : 命令無効化制御回路 7 0 8 は、条件付き実行ステータス 7 0 9 が “x’ 10” から “x’ 01” に更新する。いま、排他的条件は “1” であり、条

件付き実行ステータスが“x'01”であるので、図17（a）に従い、命令無効化信号707が生成され、“0”から“1”に更新される。

【0221】

IFステージ：命令3 [ADD R3,R6]

タイミングt31：命令3がROM717から読み出され、命令ラッチ714に格納される。

【0222】

（タイミングt4）

EXステージ：命令2 [MOV R2,R3]

タイミングt41：命令無効化信号707が“1”となっているので命令2は無効化され実行されない。つまり、この例では後続する命令2が無効化された。

【0223】

DECステージ：命令3 [ADD R3,R6]

タイミングt41：命令ラッチ714に格納された命令3が命令解読器705で解読され、R3レジスタから値が読出されD1ラッチ710に格納される。R6レジスタから値が読み出されD2ラッチ711に格納される。

【0224】

タイミングt42：命令無効化制御回路708は、条件付き実行ステータス709を“x'01”から“x'00”に更新する。いま、排他的条件は“1”であり、条件付き実行ステータスが“x'00”であるので、図17（a）に従い、命令無効化信号707は“0”となる。

【0225】

IFステージ：命令4 [MOV R4,R6]

タイミングt41：命令4がROM717から読み出され、命令ラッチ714に格納される。

【0226】

（タイミングt5）

EXステージ：命令3 [ADD R3,R6]

タイミングt51：命令無効化信号707は“0”であるので命令3の実行を

開始する。D 1 ラッチ 7 1 0 に格納された値と D 2 ラッチ 7 1 1 に格納された値が演算器 7 0 3 で加算され R6 レジスタに格納される。この例では後続命令 (2) は無効化されたが、後続命令 (2) 以後の後続命令 (3) は実行された。

## 【 0 2 2 7 】

DEC ステージ：命令 4 [MOV R4,R7]

タイミング t 5 1：命令ラッチ 7 1 4 に格納された命令 4 が命令解読器 7 0 5 で解読され、R4 レジスタから値が読出され D 1 ラッチ 7 1 0 に格納される。

## 【 0 2 2 8 】

タイミング t 5 2：命令無効化制御回路 7 0 8 は、条件付き実行ステータス 7 0 9 を “x'00” で保持する。

## 【 0 2 2 9 】

IF ステージ：命令 5 [NOP]

タイミング t 5 1：命令 5 が ROM 5 から読出され、命令ラッチ 7 1 4 に格納される。

## 【 0 2 3 0 】

以上のように本実施の形態によれば、命令 1 [SUBIFXZF R0,R1] により条件としてゼロフラグを選択し、条件付き実行ステータス 7 0 9 をセットすることで条件付き実行されるので、後続の命令には特別な条件指定フィールドを付加する必要が無く、後続命令の実行、無効化を指定することができる。

## 【 0 2 3 1 】

なお、上記の説明では R0 レジスタと R1 レジスタの値を同値として、ゼロフラグの値が “1” となる場合を示したが、R0 レジスタと R1 レジスタの値が同値ではなく “0” となる場合も、命令無効化制御回路 7 0 8 は図 1 7 (a) に従って命令無効化信号 7 0 7 を制御することにより命令無効化制御を実行することができる。この場合は後続命令 (2) が無効化されずに実行される。

## 【 0 2 3 2 】

(実施形態 6)

実施形態 6 として、本発明の第 3 および第 4 のプログラム変換装置の例を示す。実施形態 6 のプログラム変換装置を図 1 8、図 2 0、図 2 1、図 2 2 によって

説明する。なお、既に説明した図には同一番号を付して説明を省略する。

#### 【 0 2 3 3 】

本実施形態 6 のプログラム変換装置の全体構成例を図 2 0 に示す。図 2 0 に示すように本実施形態 6 のプログラム変換装置は、実施形態 2 など示した図 6 に比べ、第 2 の判定手段を備えていない構成となっている。他の部分は同様で良く、ここでの説明は省略する。

#### 【 0 2 3 4 】

本実施形態 6 のプログラム変換装置によるプログラムの書き換え動作の例を説明する。図 2 1 の書き換え前のアセンブリ言語プログラムを図 1 8 の機械語プログラムに書き換える例を示す。図 1 8 の機械語プログラムは、実施形態 5 で用いたプログラムと同様である。図 1 8、図 2 1 のニーモニック表記した命令は既に説明したプログラムと ADD 命令、MOV 命令などで使用しているオペランド（即値、レジスタ）が異なるだけであるので説明を省略する。

#### 【 0 2 3 5 】

図 2 2 は、本実施形態 6 におけるプログラム変換装置による第 2 の処理 2 0 7 の前半部を示す図である。この第 2 の処理 2 0 7 は、第 1 の処理 2 0 6 において検出された条件・判断命令を、演算命令と、該演算命令に後続し該演算命令に伴う後続命令の命令列に書き換える命令書き換え手段を実行する処理である。この第 2 の処理 2 0 7 は、第 1 の判定手段 2 0 8 と命令検索手段 2 1 0 と命令書き換え手段 2 0 5 によって実現される。

#### 【 0 2 3 6 】

なお、図 2 2 の 2 1 1 b は図 1 0 の 2 1 1 に該当する部分を示しており、それ以外は図 1 0 と同様であるとして、ここでは説明を省略する。

#### 【 0 2 3 7 】

ステップ 2 4 2 b は、第 1 の判定手段 2 0 8 で実行され、配列 2 から J + 1 の命令番号を検索する操作である。つまり、ステップ 2 4 2 b は、第 1 の命令検出手段により検出された命令に後続して、第 2 の命令検出手段により検出された命令があるかを判定する第 1 の判定手段を実行する。

#### 【 0 2 3 8 】

ステップ 2 4 3 b は命令番号に相当する命令を読み出し記憶する操作である（以下、命令 A、命令 B）。

【 0 2 3 9 】

ステップ 2 4 4 b は、命令 B の条件を条件 1 として記憶する。

【 0 2 4 0 】

ステップ 2 4 5 b は、命令検索手段 2 1 0 において、命令 A と同じ処理を実行可能で、かつ条件 1 を後続に配置可能で、かつ後続の命令を対象とする指示命令が存在するか判定する操作である。つまり、ステップ 2 4 5 b は、機械語命令コードの中から、第 1 の命令検出手段により検出された命令と同一の処理実行を示し、かつ、後続の条件付き命令の実行条件を示す指示命令を検索する指示命令検索手段を実行する処理である。

【 0 2 4 1 】

以上が本実施形態 6 の 2 1 1 b の処理である。

【 0 2 4 2 】

以下、図 2 1 の書き換え前アセンブリ言語プログラムを図 1 8 の機械語プログラムに書き換える動作を第 1 の処理 2 0 6 と第 2 の処理 2 0 7 の順番に説明する。なお、1 行目の命令を命令（1）、2 行目の命令を命令（2）のように括弧で以下表現する。

【 0 2 4 3 】

（第 1 の処理 2 0 6）

実施形態 2 と同様に処理され、最終的に、配列 1 には 1、3 が、配列 2 には 2 が格納されており、それらは第 2 の処理 2 0 7 で使用される。

【 0 2 4 4 】

（第 2 の処理 2 0 7）

ステップ 2 4 0 : 配列 1 の 1 番目の要素から処理を開始する。

【 0 2 4 5 】

ステップ 2 4 1 : 配列 1 から 1 番目の命令番号である 1 を読み出し変数 J に格納する。

【 0 2 4 6 】



ステップ 2 4 2 b : 第 1 の判定手段 2 0 8 において、配列 2 から J + 1 の命令番号を検索する。いま J = 1 であり、命令 ( 2 ) が存在するのでステップ 2 4 3 b に移る。

【 0 2 4 7 】

ステップ 2 4 3 b : 命令列記憶装置から変数 J から J + 1 に対応する命令を読み出し、それぞれ命令 A (SUB)、命令 B (MOVIFNZF) として記憶する。

【 0 2 4 8 】

ステップ 2 4 4 b : 命令 B の条件を条件 1 として、次のステップに移る。

【 0 2 4 9 】

ステップ 2 4 5 b : 命令検索手段 2 1 0 において、命令 A と同じ処理を実行可能で、かつ条件 1 を後続に配置可能で、かつ後続の命令を対象とする指示命令が存在するか判定する。ここでは SUBIFXZF が候補として存在するので、ステップ 2 4 6 に移る。

【 0 2 5 0 】

ステップ 2 4 6 : 命令書き換え手段 2 0 5 において、命令 A を SUBIFXZF に置き換える。また命令 B に対応する処理を無条件に実行する命令に置き換える。次にステップ 2 4 7 に移る。

【 0 2 5 1 】

ステップ 2 4 7 : 命令書き換え手段 2 0 5 において、命令列記憶装置の変数 J から J + 1 に対応する命令を、それぞれ命令 A (SUBIFXZF)、命令 B (MOV) に書き換える。

【 0 2 5 2 】

ステップ 2 4 8 : 配列 1 のすべての命令番号を処理したか判定する。まだ残っているのでステップ 2 4 9 に移る。

【 0 2 5 3 】

ステップ 2 4 9 : 配列 1 における次の命令番号を処理するように、配列 1 の要素を 1 増分する。

【 0 2 5 4 】

ステップ 2 4 1 : 配列 1 から 2 番目の命令番号である 3 を読み出し変数 J に格

納する。

【0255】

ステップ242：第1の命令検出手段202において、配列2から変数J+1の命令番号を検索する。いまJ=3であり、配列2に命令(4)が存在するか否かを調べる。ここでは存在しないので、次にステップ248に移る。

【0256】

ステップ248：配列1のすべての命令番号を処理したか判定する。配列1にもう命令番号は存在しないのでステップ250に移る。

【0257】

ステップ250：第2の処理207を完了する。

【0258】

以上のように、本実施形態6によれば、命令列記憶装置201に格納された図21の命令(1)と命令(2)が、図18の命令(1)と命令(2)に書き換えられる。このように、プログラム変換装置62は、命令列記憶装置201に最終的に格納される命令列を、実行するための条件を明示した指示命令と、実行するための条件が明示されない命令とからなる命令列で書き換えることが可能である。

【0259】

なお、本実施形態6のプログラム変換装置にはアセンブリ言語プログラムを入力するものとしたが、機械語プログラムを入力として新たな機械語プログラムを生成するプログラム変換装置としてもよい。

【0260】

(実施形態7)

実施形態7として、本発明の第3および第4のデータ処理装置の例を示す。本実施形態7は、条件付き実行ステータス709の更新を操作することで、実行するための条件が明示されない複数からなる命令を条件付き実行するものである。本実施形態5のデータ処理装置では後続命令が1つの命令であったが、本実施形態7は、後続命令が複数の命令列からなる後続命令群である。

【0261】

以下、本発明の実施形態 7 のデータ処理装置について図 1、図 2、図 2 3、図 2 4、図 2 5 を用いて説明する。なお、実施形態 1 などと同様の部分については、同一図と符号を付与して説明を省略する。

#### 【0 2 6 2】

本実施形態 7 のデータ処理装置の全体構成例に関しては図 1、プロセッサの構成例に関しては図 2 で良く、実施形態 1 と同様であるのでここでの説明は省略する。

#### 【0 2 6 3】

本実施形態 7 における命令無効化制御方法と条件付き実行ステータス 7 0 9 の遷移について図 2 3 を用いて説明する。図 2 3 (a) は、命令無効化制御回路 7 0 8 が EX ステージの処理を無効化するか表している (命令無効化信号 7 0 7)。同図の排他的条件 7 0 1 とは、本実施形態の指示命令によって選択されたプロセッサステータスの状態であり “0” か “1” の 2 値をとる。同図の条件付き実行ステータス 7 0 9 とは、同じく本実施形態の指示命令によって後続の命令を条件付き実行している状態を表している。図 2 3 (a) に従えば、例えば、排他的条件 7 0 1 が “1” で条件付き実行ステータス 7 0 9 が “x’01” および “x’10” の両ケースの場合は EX ステージの処理が無効化される。図 2 3 (b) は、条件付き実行ステータス 7 0 9 の遷移図であり、シーケンサ 7 1 6 により制御される。条件付き実行ステータス 7 0 9 は、初期値 “x’00” から前記の指示命令が解読された後 “x’11” にセットされ、次命令が解読されると “x’1” デクリメントしてゆき、“x’00” まで順に更新される。

#### 【0 2 6 4】

以上が本実施形態 7 のデータ処理装置の構成例である。以下に本実施形態 7 のデータ処理装置の動作の例を説明する。一例として図 2 4 に示す機械語プログラム 4 を実行処理する場合の動作を説明する。図 2 4 は機械語プログラム 4 を表しており、ニーモニック表記である。その命令について簡単な説明を付する。

#### 【0 2 6 5】

(機械語プログラム 4)

1 行目: SUBIFNXZF R0,R1 (R0 レジスタの値から R1 のレジスタの値を減算して

結果をR1レジスタに格納する。さらに後続の2命令をプロセッサステータスのゼロフラグが0のとき実行する条件付き実行命令として指示する)

2行目: MOV R2,R3 (R2レジスタの値をR3レジスタに格納する。)

3行目: MOV R5,R4 (R5レジスタの値をR4レジスタに格納する。)

4行目: ADD R3,R6 (R3レジスタの値とR6レジスタの値を加算して結果をR6レジスタに格納する)

5行目: MOV R4,R7 (R4レジスタの値をR7レジスタに格納する)

このように、機械語プログラム4において、1行目の指示命令SUBIFNXZFは、後続する2行目から3行目に配置された命令群を実行するかしないかを定める条件付き命令となっている。指示命令SUBIFNXZFの演算結果に基づいてゼロフラグの状態が決定され、当該ゼロフラグの状態に応じて後続の命令群が実行されるか否かが決まる。

#### 【0266】

図25はプロセッサ64の動作タイミング図を示すものである。また、図25に示すようにステージの前半(例えばタイミングt11)と後半(タイミングt12)を表すタイミングにより命令無効化制御回路708は動作するものとする。特にDECステージではタイミングの後半で命令無効化制御回路708により条件付き実行ステータス709を更新するものとする。ここで、時間が経過する順にタイミング毎に図24の機械語プログラムを実行する場合の動作を説明する。以下、時間が経過する順にタイミング毎にその動作を説明する。なお、R0とR1には同値である“x'0001”が格納され、R2には“x'0002”、R3には“x'0003”、R4には“x'0004”、R5には“x'0005”が格納されていたとする。

#### 【0267】

(タイミングt1)

IFステージ: 命令1 [SUBIFNXZF R0,R1]

タイミングt11: 命令1がROM717から読み出され、命令ラッチ714に格納される。

#### 【0268】

(タイミングt2)

DECステージ：命令 1 [SUBIFNXZF R0,R1]

タイミング t 2 1：命令ラッチ 7 1 4 に格納された命令 1 が命令解読器 7 0 5 で解読され、R0レジスタから値が読出されD 1 ラッチ 7 1 0 に “x’ 0001” が格納される。R1レジスタから値が読み出されD 2 ラッチ 7 1 1 に “x’ 0001” が格納される。

【 0 2 6 9 】

タイミング t 2 2：命令無効化制御回路 7 0 8 は、条件付き実行ステータス 7 0 9 を “x’ 00” から “x’ 11” にセットする。排他的条件 7 0 1 の値を表わすレジスタとしてゼロフラグがセレクタ 7 0 2 により選択される。

【 0 2 7 0 】

IFステージ：命令 2 [MOV R2,R3]

タイミング t 2 1：命令 2 がROM 7 1 7 から読み出され、命令ラッチ 7 1 4 に格納される。

【 0 2 7 1 】

(タイミング t 3)

EXステージ：命令 1 [SUBIFNXZF R0,R1]

タイミング t 3 1：命令無効化信号 7 0 7 は初期状態のまま “0” となっており、EXステージの実行が行われ、命令 1 の実行が開始される。D 1 ラッチ 7 1 0 とD 2 ラッチ 7 1 1 に格納された値が演算器 7 0 3 で減算される。ここではD 1 ラッチ 7 1 0 とD 2 ラッチ 7 1 1 には同値 “x’ 0001” が格納されているので、減算結果としてR1レジスタに “x’ 0000” が格納される。この結果ゼロフラグには “1” が格納される。

【 0 2 7 2 】

DECステージ：命令 2 [MOV R2,R3]

タイミング t 3 1：命令ラッチ 7 1 4 に格納された命令 2 が命令解読器 7 0 5 で解読され、R2レジスタから値が読み出されD 1 ラッチ 7 1 0 に “x’ 0002” が格納される。

【 0 2 7 3 】

タイミング t 3 2：命令無効化制御回路 7 0 8 は、条件付き実行ステータス 7

09が“x'11”から“x'10”に更新する。いま、排他的条件は“1”であり、条件付き実行ステータスが“x'10”であるので、図23（a）に従い、命令無効化信号707が生成され“0”から“1”に更新される。

## 【0274】

IFステージ：命令3 [MOV R5,R4]

タイミングt31：命令3がROM717から読み出され、命令ラッチ714に格納される。

## 【0275】

（タイミングt4）

EXステージ：命令2 [MOV R2,R3]

タイミングt41：命令無効化信号707が“1”であるので命令2は無効化され実行されない。つまり、この例では後続命令のうち命令2は無効化された。

## 【0276】

DECステージ：命令3 [MOV R5,R4]

タイミングt41：命令ラッチ714に格納された命令3が命令解読器705で解読され、R5レジスタから値が読み出されD1ラッチ710に“x'0005”が格納される。

## 【0277】

タイミングt42：命令無効化制御回路708は、条件付き実行ステータス709を“x'10”から“x'01”に更新する。いま、排他的条件は“1”であり、条件付き実行ステータスが“x'01”であるので、図23（a）に従い、命令無効化信号707が生成され“1”が維持される。

## 【0278】

IFステージ：命令4 [ADD R3,R6]

タイミングt41：命令4がROM717から読み出され、命令ラッチ714に格納される。

## 【0279】

（タイミングt5）

EXステージ：命令3 [MOV R5,R4]

タイミング t 5 1 : 命令無効化信号 7 0 7 が “ 1 ” であるので命令 3 は無効化される。つまり、この例では後続命令のうち命令 2 と命令 3 が無効化された。

## 【 0 2 8 0 】

DECステージ：命令 4 [ADD R3,R6]

タイミング t 5 1 : 命令ラッチ 7 1 4 に格納された命令 4 が命令解読器 7 0 5 で解読され、R3レジスタから値が読み出され D 1 ラッチ 7 1 0 に格納される。R6レジスタから値が読み出され D 2 ラッチ 7 1 1 に格納される。

## 【 0 2 8 1 】

タイミング t 5 2 : 命令無効化制御回路 7 0 8 は、条件付き実行ステータス 7 0 9 を “ x' 01 ” から “ x' 00 ” に更新する。いま、排他的条件は “ 1 ” であり、条件付き実行ステータスが “ x' 00 ” であるので、図 2 3 ( a ) に従い、命令無効化信号 7 0 7 は生成されず “ 1 ” から “ 0 ” に更新される。

## 【 0 2 8 2 】

IFステージ：命令 5 [MOV R4,R7]

タイミング t 5 1 : 命令 5 が R O M 7 1 7 から読み出され、命令ラッチ 7 1 4 に格納される。

## 【 0 2 8 3 】

(タイミング t 6 )

EXステージ：命令 4 [ADD R3,R6]

タイミング t 6 1 : 命令無効化信号 7 0 7 が “ 0 ” であるので命令 4 は実行される。D 1 ラッチ 7 1 0 に格納された値と D 2 ラッチ 7 1 1 に格納された値が演算器 7 0 3 で加算され R6レジスタに格納される。

## 【 0 2 8 4 】

DECステージ：命令 5 [MOV R4,R7]

タイミング t 6 1 : 命令ラッチ 7 1 4 に格納された命令 5 が命令解読器 7 0 5 で解読され、R4レジスタから値が読み出され D 1 ラッチ 7 1 0 に格納される。

## 【 0 2 8 5 】

タイミング t 6 2 : 命令無効化制御回路 7 0 8 は、条件付き実行ステータス 7 0 9 を “ x' 00 ” で保持する。

【 0 2 8 6 】

IFステージ：命令 6 [NOP]

タイミング t 6 1：命令 6 が ROM 7 1 7 から読み出され、命令ラッチ 7 1 4 に格納される。

【 0 2 8 7 】

以上のように本実施の形態によれば、命令 1 [SUBIFNXZF R0,R1]により条件としてゼロフラグを選択し、条件付き実行ステータス 7 0 9 をセットすることで条件付き実行されるので、後続の命令群には条件指定フィールドを付加する必要が無く、後続命令群の実行、無効化を指定することができる。

【 0 2 8 8 】

なお、上記の説明では R0 レジスタと R1 レジスタの値を同値として、ゼロフラグの値が“1”となる場合を示したが、R0 レジスタと R1 レジスタの値が同値ではなく“0”となる場合も、命令無効化制御回路 7 0 8 は図 2 3 (a) に従って命令無効化信号 7 0 7 を制御することにより命令無効化制御を実行することができる。この場合は後続命令のうち命令 2 と命令 3 の両方が実行される。

【 0 2 8 9 】

また、上記に示した例では、命令 1 はプロセッサステータスワード 7 1 2 のうちゼロフラグを命令無効化信号 7 0 7 として選択する命令としたが、限定されるものではなく、他のプロセッサステータスワード 7 1 2 を選択しても良い。

【 0 2 9 0 】

(実施形態 8)

実施形態 8 として、本発明の第 1 および第 2 のプログラム変換装置の例を示す。実施形態 6 では後続命令が 1 つの命令であったが、本実施形態 8 は、後続命令が複数の命令列からなる後続命令群である。

【 0 2 9 1 】

本発明のプログラム変換装置を図 2 0、図 2 4、図 2 6、図 2 7 によって説明する。既に説明した図には同一番号を付して説明を省略する。

【 0 2 9 2 】

本実施形態 8 のデータ処理装置の全体構成例に関しては図 2 0 で良く、実施形



態6と同様であるのでここでの説明は省略する。

【0293】

本実施形態4のプログラム変換装置によるプログラムの書き換え動作の例を説明する。図26の書き換え前のアセンブリ言語プログラムを図24の機械語プログラムに書き換える例を示す。図24の機械語プログラムは、実施形態7で用いたプログラムと同様である。図24、図26のニーモニック表記した命令はすでに説明しており、ADD命令、MOV命令などで使用しているオペランド（即値、レジスタ）が異なるだけであるのでここでは説明を省略する。

【0294】

図27は、本実施形態8におけるプログラム変換装置による第2の処理207の前半部を示す図である。この第2の処理207は、第1の処理206において検出された条件・判断命令を、演算命令と、該演算命令に後続し該演算命令に伴う後続命令群の命令列に書き換える命令書き換え手段を実行する処理である。この第2の処理207は、第1の判定手段208と命令検索手段210と命令書き換え手段205によって実現される。

【0295】

なお、図27の211cは図10の211に該当する部分を示しており、それ以外は図10と同様であるとして、ここでは説明を省略する。

【0296】

ステップ242cは、第1の判定手段208で実行され、配列2からJ+1、J+2の命令番号を検索する操作である。つまり、ステップ242cは、第1の命令検出手段により検出された命令に後続して、第2の命令検出手段により検出された命令群があるかを判定する第1の判定手段を実行する。

【0297】

ステップ243cは、命令番号に相当する命令を読み出し記憶する操作である（以下、命令A、命令B、命令C）。

【0298】

ステップ244cは、命令Bの条件と命令Cの条件が同一の条件1であるか判定する操作である。

## 【0299】

ステップ245cは、命令Aと同じ処理を実行可能で、かつ条件1を後続に配置可能で、かつ後続の2命令を対象とする指示命令が存在するか判定する操作である。つまり、ステップ245cは、機械語命令コードの中から、第1の命令検出手段により検出された命令と同一の処理実行を示し、かつ、後続の条件付き命令群の実行条件を示す指示命令を検索する指示命令検索手段を実行する処理である。

## 【0300】

以上が本実施形態8の211cの処理である。

## 【0301】

以下、図26の書き換え前アセンブリ言語プログラムを図24の機械語プログラムに書き換える動作を第1の処理206と第2の処理207の順番に説明する。なお、1行目の命令を命令(1)、2行目の命令を命令(2)のように括弧で以下表現する。

## 【0302】

(第1の処理206)

実施形態1と同様に処理され、最終的に、配列1には1、4が、配列2には2、3が格納されており、それらは第2の処理207で使用される。

## 【0303】

(第2の処理207)

ステップ240：配列1の1番目の要素から処理を開始する。

## 【0304】

ステップ241：配列1から1番目の命令番号である1を読み出し変数Jに格納する。

## 【0305】

ステップ242c：第1の判定手段208において、配列2からJ+1、J+2の命令番号を検索する。いまJ=1であり、命令(2)および命令(3)は存在するのでステップ243cに移る。

## 【0306】

ステップ 243c : 命令列記憶装置から変数 J から J + 2 に対応する命令を読み出し、それぞれ命令 A (SUB)、命令 B (MOVIFNZF)、命令 C (MOVIFNZF) として記憶する。

【0307】

ステップ 244c : 命令 B の条件と命令 C の条件が同一の条件であるか判定する。ここでは同一であるのでそれを条件 1 として次の操作に移る。

【0308】

ステップ 245c : 命令検索手段 210 において、命令 A と同じ処理を実行可能で、かつ条件 1 を後続に配置可能で、かつ後続の 2 つの命令を対象とする指示命令が存在するか判定する。ここでは SUBIFNXZF が候補として存在するので、ステップ 246 に移る。

【0309】

ステップ 246 : 命令書き換え手段 205 において、命令 A を SUBIFNXZF に置き換える。また命令 B と命令 C に対応する処理を無条件に実行する命令に置き換える。次にステップ 247 に移る。

【0310】

ステップ 247 : 命令書き換え手段 205 において、命令列記憶装置の変数 J から J + 2 に対応する命令を、それぞれ命令 A (SUBIFXZF)、命令 B (MOV)、命令 C (MOV) に書き換える。

【0311】

ステップ 248 : 配列 1 のすべての命令番号を処理したか判定する。まだ残っているのでステップ 249 に移る。

【0312】

ステップ 249 : 配列 1 における次の命令番号を処理するように、配列 1 の要素を 1 増分する。

【0313】

ステップ 241 : 配列 1 から 2 番目の命令番号である 4 を読み出し変数 J に格納する。

【0314】

ステップ 2 4 2 : 第 1 の命令検出手段 2 0 2 において、配列 2 から変数  $J + 1$ 、 $J + 2$  の命令番号を検索する。いま  $J = 4$  であり、配列 2 に命令 (5) および命令 (6) が存在するか否かを調べる。ここでは存在しないので、次にステップ 2 4 8 に移る。

【 0 3 1 5 】

ステップ 2 4 8 : 配列 1 のすべての命令番号を処理したか判定する。配列 1 にもう命令番号は存在しないのでステップ 2 5 0 に移る。

【 0 3 1 6 】

ステップ 2 5 0 : 第 2 の処理 2 0 7 を完了する。

【 0 3 1 7 】

以上のように、本実施形態 8 によれば、命令列記憶装置 2 0 1 に格納された図 2 6 の命令 (1) と命令 (2) と命令 (3) が、図 2 4 の命令 (1) と命令 (2) と命令 (3) に書き換えられる。

【 0 3 1 8 】

以上、説明したように、プログラム変換装置 6 2 は、命令列記憶装置 2 0 1 に最終的に格納される命令群を、実行するための条件を明示した指示命令と、実行するための条件が明示されない命令群とからなる命令列で書き換えることが可能である。上記の手法をとれば、従来の技術と比べ、多種多様な無条件実行命令を、新たにコード割付けすることなく、プロセッサの状態に応じて条件付き実行する命令として処理することが可能となる。

【 0 3 1 9 】

なお、本実施形態 8 のプログラム変換装置にはアセンブリ言語プログラムを入力するものとしたが、機械語プログラムを入力として新たな機械語プログラムを生成するプログラム変換装置としてもよい。

【 0 3 2 0 】

(実施形態 9)

本発明の実施形態 9 のデータ処理装置は、命令をフェッチ、解読、実行からなる少なくとも 3 段以上のパイプラインによって処理するデータ処理装置であって、命令を先読みし記憶しておく命令先読み手段と、命令を解読する命令解読手段

と、命令を実行する命令実行手段と、後方に配置された命令あるいは命令列が無効化されるとき前記命令先読み手段から無効化すべき命令をスキップするものである。実施形態 1、3、5、7 のデータ処理装置では、無効化される命令は EX ステージにおいて実行されないという方式を採っているが、実施形態 9 のデータ処理装置は、無効化される命令は IF ステージにおいてその読み込みがスキップされるものである。

### 【 0 3 2 1 】

実施形態 9 のデータ処理装置を図 1、図 2、図 28、図 29、図 30、図 31 によって説明する。なお、実施形態 1 と同様の部分については、同一図と符号を付与して説明を省略する。

### 【 0 3 2 2 】

本実施形態 9 のデータ処理装置の全体構成例に関しては図 1、プロセッサの構成例に関しては図 2 で良く、実施形態 1 と同様であるのでここでの説明は省略する。

### 【 0 3 2 3 】

図 28 は命令フェッチ回路 706 の構成例を示す図である。141 は ROM 717 からの命令である。142 は命令先読み制御信号 713 に応じてバッファ 0 とバッファ 1 に格納された命令を選択する命令セクタである。143 は命令解読の対象となる命令である。本実施形態 9 における命令フェッチ回路 706 は一般的な命令キューにより構成されており、ROM 717 からの命令 141 を先読みして記憶するための 5 つのバッファ 0 ～バッファ 4 には、パイプラインを制御するタイミングに連動して命令が取り込まれる。先読みされた命令は、常に番号の小さいバッファ 0 を先頭に記憶されるものとする。もし 4 つの命令が記憶されているときにはバッファ 0 からバッファ 3 に命令は格納されていることとなる。また、命令フェッチ回路 706 は、IF ステージで解読器に必要な命令が存在しない場合には命令をフェッチするために他のステージ (DEC、EX ステージ) の処理を停止させ、他のステージ (DEC、EX ステージ) の処理が伸張 (数ステージを要する) された場合には命令を命令キューが埋まるまで先読みするものとする。なお、実施の形態 1 の命令ラッチ 714 は、本実施の形態ではバッファ 0 とバッファ

ァ 1 と命令セクタ 1 4 2 により構成されている。

【 0 3 2 4 】

ここで、本実施形態 9 における命令フェッチ回路 7 0 6 の制御方法について図 2 9 により説明する。命令キューの動作は 2 通りあり命令先読み制御信号 7 1 3 により選択され切り替えられる。

【 0 3 2 5 】

第 1 の読み方は、図 2 9 ( a ) に示したもので、ROM 7 1 7 からの命令はバッファ 4 に、バッファ 4 に記憶された命令はバッファ 3 へ、バッファ 3 に記憶された命令はバッファ 2 へ、バッファ 2 に記憶された命令はバッファ 1 へ、バッファ 1 に記憶された命令はバッファ 0 へ、バッファ 0 に記憶された命令は解読器へ転送されるものである。この読み方によれば、命令はシーケンス通りに順に読み込まれて行くこととなる。

【 0 3 2 6 】

第 2 の読み方は、図 2 9 ( b ) に示したもので、ROM 7 1 7 からの命令はバッファ 3 へ、バッファ 4 に記憶された命令はバッファ 2 へ、バッファ 3 に記憶された命令はバッファ 1 へ、バッファ 2 に記憶された命令はバッファ 0 へ、バッファ 1 に記憶された命令は解読器へ転送されるものである。この読み方によれば、命令は図 2 9 ( a ) に示したバッファよりも時系列的に一つ進んだバッファに転送され、バッファ 1 に格納された命令が解読器に転送されることとなり、一つ分の命令がスキップされた効果が生じる。

【 0 3 2 7 】

つまり、命令先読み制御信号 7 1 3 によりバッファ 0 かバッファ 1 の命令を選択して命令解読の対象となる命令 1 4 3 として解読器へ転送することができる。つまり、命令セクタ 1 4 2 は、命令先読み制御信号 7 1 3 が “ 0 ” のとき図 2 9 ( a ) を、“ 1 ” のとき図 2 9 ( b ) を選択するものとする。

【 0 3 2 8 】

ここで、本実施形態 9 における命令無効化制御方法と条件付き実行ステータス 7 0 9 の遷移図を図 3 0 に示す。図 3 0 の排他的条件 7 0 1 とは、本発明の指示命令によって選択されたプロセッサステータスの状態であり、“ 0 ” か “ 1 ” の

2 値をとる。図 3 0 の条件付き実行ステータス 7 0 9 とは、同じく本発明の指示命令によって後続の命令を条件付き実行している状態を表している。図 3 0 ( a ) は、命令無効化制御回路 7 0 8 が EX ステージの処理を無効化するか実行するかを表している。ここで、図 3 ( a ) と異なり、排他的条件 7 0 1 が “ 1 ” で条件付き実行ステータス 7 0 9 が “ x' 01 ” の場合は存在しない。

#### 【 0 3 2 9 】

図 3 0 ( b ) は、条件付き実行ステータス 7 0 9 の遷移図であり、シーケンサ 7 1 6 により制御される。条件付き実行ステータス 7 0 9 は、指示命令が解読された時点で “ x' 00 ” から “ x' 11 ” にセットされ、次の DEC ステージで後続の命令が解読され “ x' 10 ” に更新される。次にシーケンサ 7 1 6 は、排他的条件 7 0 1 が “ 1 ” ならば “ x' 10 ” から “ x' 00 ” に、排他的条件 7 0 1 が “ 0 ” ならば “ x' 10 ” から “ x' 01 ” に条件付き実行ステータス 7 0 9 を更新する。つまり、シーケンサ 7 1 6 は条件付き実行ステータス 7 0 9 が “ x' 10 ” のときのみ、排他的条件 7 0 1 に応じて次の値を更新する。

#### 【 0 3 3 0 】

図 3 0 ( c ) は、命令先読み制御信号 7 1 3 の状態表を示す図である。本実施形態 9 では、排他的条件と条件付き実行ステータスの状態に応じて命令先読み制御信号 7 1 3 の状態が決められる。命令先読み制御信号 7 1 3 が切り替えられることにより先読み方法が図 2 9 ( a ) となるか図 2 9 ( b ) が決まる。図 3 0 ( c ) に見るように、排他的条件 7 0 1 が “ 1 ” かつ条件付き実行ステータス 7 0 9 が “ x' 10 ” のときのみ命令先読み制御信号 7 1 3 が “ 1 ” となる。

#### 【 0 3 3 1 】

以上のように構成された本実施形態 9 によるデータ処理装置について、以下その動作を説明する。なお、以下の説明では実施形態 1 で用いた、図 5 の機械語プログラム 1 を実行させる。

#### 【 0 3 3 2 】

図 3 1 は、プロセッサ 6 4 の動作タイミング図を示すものである。図 3 1 は、図 5 を実行する場合の動作をパイプラインの IF ステージ、DEC ステージ、EX ステージで処理される命令をマシンサイクルと呼ばれるタイミング毎に示している。

また、図 3 1 に示すようにステージの前半（例えばタイミング t 1 1）と後半（タイミング t 1 2）を表すタイミングにより命令無効化制御回路 7 0 8 は動作するものとする。特に DEC ステージではタイミングの後半で命令無効化制御回路 7 0 8 により条件付き実行ステータス 7 0 9 と命令先読み制御信号 7 1 3 を更新するものとする。また、EX ステージではタイミングの前半で命令無効化信号 7 0 7 に応じて、このステージで命令を実行するか、命令を無効化するか決定されるものとする。また、IF ステージと DEC ステージでは、タイミングの前半で命令先読み制御信号 7 1 3 に応じて、命令フェッチ回路 7 0 6 から命令を読み込むものとする。以下、時間が経過する順にタイミング毎にその動作を説明する。なお、R0 と R1 には同値である “x’ 0001” が格納され、R2 には “x’ 0002”、R3 には “x’ 0003”、R4 には “x’ 0004”、R5 には “x’ 0005” が格納されていたとする。

## 【 0 3 3 3 】

（タイミング t 1）

IF ステージ：命令 1 [SUBIFEZF R0,R1]

タイミング t 1 1：命令先読み制御信号 7 1 3 が “0” であるので、命令 1 がバッファ 1 からバッファ 0 に、命令 2 がバッファ 2 からバッファ 1 に、それぞれ格納される。

## 【 0 3 3 4 】

（タイミング t 2）

DEC ステージ：命令 1 [SUBIFEZF R0,R1]

タイミング t 2 1：命令先読み制御信号 7 1 3 が “0” であるので、命令セクタ 1 4 2 によりタイミング t 1 でバッファ 0 に格納された命令 1 が命令解読器 7 0 5 に転送され解読される。R0 レジスタから値が読み出され D 1 ラッチ 7 1 0 に “x’ 0001” が格納される。R1 レジスタから値が読み出され D 2 ラッチ 7 1 1 に x’ 0001 が格納される。

## 【 0 3 3 5 】

タイミング t 2 2：命令無効化制御回路 7 0 8 は、条件付き実行ステータス 7 0 9 を “x’ 11” にセットする。ゼロフラグが排他的条件 7 0 1 としてセクタ 7 0 2 により選択される。



【0336】

IFステージ：命令2 [MOV R2,R4]

タイミングt 2 1：命令先読み制御信号7 1 3が“0”であるので、命令2がバッファ1からバッファ0に、命令3がバッファ2からバッファ1に、それぞれ格納される。

【0337】

(タイミングt 3)

EXステージ：命令1 [SUBIFEZF R0,R1]

タイミングt 3 1：命令無効化信号7 0 7は初期状態のまま“0”であるので命令1の実行を開始する。D 1ラッチ7 1 0とD 2ラッチ7 1 1に格納された値が演算器7 0 3で減算される。ここではD 1ラッチ7 1 0とD 2ラッチ7 1 1には同値“x' 0001”が格納されているので、減算結果としてR1レジスタ“x' 0000”が格納される。この結果、ゼロフラグには“1”が格納される。

【0338】

DECステージ：命令2 [MOV R2,R4]

タイミングt 3 1：命令先読み制御信号7 1 3が“0”であるので、命令セクタ1 4 2によりタイミングt 2のバッファ0から命令2が命令解読器7 0 5に転送され解読される。R2レジスタから値が読み出されD 1ラッチ7 1 0に“x' 00 02”が格納される。

【0339】

タイミングt 3 2：命令無効化制御回路7 0 8は、条件付き実行ステータス7 0 9を“x' 11”から“x' 10”に更新する。ここで、条件付き実行ステータス7 0 9が“x' 10”で排他的条件7 0 1が“1”であるので命令先読み制御信号7 1 3は“0”から“1”に更新される。

【0340】

IFステージ：命令3 [MOV R3,R4]

タイミングt 3 1：命令先読み制御信号7 1 3が“0”であるので、図29 (a)の先読み方法に従い、命令3がバッファ1からバッファ0に、命令4がバッファ2からバッファ1に、それぞれ格納される。

【0341】

(タイミング t 4)

EXステージ：命令 2 [MOV R2,R4]

タイミング t 4 1：命令無効化信号 707 は“0”であるので命令 2 の実行を開始する。D1 ラッチ 710 に格納された値が演算器 703 をスルーして R4 レジスタに格納される。

【0342】

DECステージ：命令 4 [ADD R5,R6]

タイミング t 4 1：命令先読み制御信号 713 が“1”であるので、図 29 (b) の先読み方法に従い、命令セクタ 142 によりタイミング t 3 のバッファ 1 から命令 4 が命令解読器 705 に転送され解読される。R5 レジスタから値が読み出され D1 ラッチ 710 に、R6 レジスタから値が読み出され D2 ラッチ 711 に格納される。

【0343】

タイミング t 4 2：命令無効化制御回路 708 は、条件付き実行ステータス 709 を“x'10”から“x'00”に更新する。図 30 (a) に従い、命令無効化信号 707 は生成されず、“0”となる。ここで、条件付き実行ステータス 709 が“x'00”で排他的条件 701 が“1”であるので命令先読み制御信号 713 は“1”から“0”に更新される。

【0344】

IFステージ：命令 5 [NOP]

タイミング t 4 1：命令先読み制御信号 713 が“1”であるので、図 29 (b) の先読み方法に従い、命令 5 がバッファ 2 からバッファ 0 に、命令 6 がバッファ 3 からバッファ 1 に、それぞれ格納される。

【0345】

(タイミング t 5)

EXステージ：命令 4 [ADD R5,R6]

タイミング t 5 1：命令無効化信号 707 は“0”であるので、命令 4 の実行を開始する。D1 ラッチ 710 と D2 ラッチ 711 の値が加算され結果が R6 レジ

スタに格納される。

【0346】

DECステージ：命令5 [NOP]

タイミングt51：命令先読み制御信号713が“0”であるので、命令セレクタ142によりタイミングt4のバッファ0から命令5が命令解読器705に転送され解読される。ここではNOP命令のため何もしない。

【0347】

タイミングt52：命令無効化制御回路708は、条件付き実行ステータス709を“x'00”、命令先読み制御信号713を“0”として保持する。

【0348】

IFステージ：命令6 [NOP]

タイミングt51：命令先読み制御信号713が“0”であるので、図29（a）の先読み方法に従い、命令6がバッファ1からバッファ0に、命令7がバッファ2からバッファ1に、それぞれ格納される。

【0349】

以上のように、先に示した実施形態1では命令3をタイミングt5で無効化して命令4をタイミングt6で実行していたが、本実施形態9では命令4をタイミングt5で実行可能となる。これはタイミングt32で命令先読み制御信号713が生成され、タイミングt42では命令無効化信号707を生成しないように命令無効化制御回路708が構成されているからである。

【0350】

つまり、本実施形態9によれば、本発明における指示命令によりセットされた条件付き実行ステータス709がDECステージで更新され、条件付き実行ステータス709と排他的条件701に基づき無効化すべき命令を確実にスキップさせるために、命令先読み制御信号713と命令無効化信号707を制御しているので、指示命令が解読され排他的条件701に応じて後方に配置された命令あるいは命令列が無効化されるときには、命令先読み手段から無効化すべき命令をスキップして前記命令解読手段に命令を転送させることができパイプラインハザードを低減できる。

## 【 0 3 5 1 】

以上の構成により、分岐命令を削減しパイプラインの乱れを低減させる条件付き実行命令を備えたデータ処理装置でも、ある二つの処理のいずれか一方を条件に従って実行させるような場合分けのプログラムでは、必ず無効化される命令によってパイプラインの乱れが生じていたが、本発明によるデータ処理装置は後方に配置された命令あるいは命令列が無効化されるときにパイプラインの乱れを生じさせること無くデータ処理ができる。

## 【 0 3 5 2 】

また、本実施形態 9 では、実施形態 1 に示した単一の命令を排他的な条件付き実行命令として実行する場合であったが、実施の形態 2 に示した条件の明示されない 2 つの命令列を排他的な条件付き実行命令として実行する場合に適用してもよい。このとき後方に配置された命令列が無効化される場合には、複数の命令（実施の形態 2 では 2 つ）が無効化の対象となるので、パイプライン・フラッシュを回避されるステージが多く、より効果的である。このとき、本実施の形態のように単一の命令をスキップするのに代えて、命令キューは複数の命令をスキップする機構を備えればよい。また、実施形態 8 に示した条件の明示されない 1 つの命令群を条件付き実行命令として指示する場合にも、命令列が無効化される場合には、指示命令のすぐ後に配置された 1 つの命令を除いた命令をスキップさせるように構成してもよい。

## 【 0 3 5 3 】

また、実施形態 9 ではパイプライン段数を 3 段としたが、MEM ステージ（メモリアクセス）、WB ステージ（レジスタ書き戻し）を加えた 5 段パイプラインであってもよく、本実施の形態によりパイプライン段数を制限するものではない。

## 【 0 3 5 4 】

## 【発明の効果】

本発明のデータ処理装置およびプログラム変換装置によれば、後続の命令列を条件付き実行命令として先行的に指示する指示命令によって無条件実行命令を条件付きとして実行できるので、データ処理装置の命令の種類や割付けの自由度を

増やしプログラムメモリの小容量化に貢献する。

【 0 3 5 5 】

さらに、ある条件に応じて二つの処理のどちらか一方を実行したり、ある条件が成立するときのみ決められた処理を実行したりする条件付き実行によるデータ処理を、無効化される処理により本来なら発生するパイプラインハザードを抑えることによって、データ処理効率の向上を図ることができるのでその実用的効果は大きい。

【図面の簡単な説明】

【図 1】 本発明のデータ処理装置の概略構成図

【図 2】 本発明のプロセッサの構成を示すブロック図

【図 3】 本発明の実施形態 1 における命令無効化制御方法と条件付き実行ステータス遷移図

【図 4】 本発明の実施形態 1 におけるプロセッサの動作タイミング図

【図 5】 本発明の実施形態 1 の指示命令を用いた機械語プログラム 1 を示す図

【図 6】 本発明のプログラム変換装置の概略構成図

【図 7】 本発明の実施形態 2 の書き換え前のアセンブリ言語プログラムを示す図

【図 8】 本発明の実施形態 2 の書き換え後の機械語プログラムを示す図

【図 9】 本発明の実施形態 2 のプログラム変換装置による第 1 の処理を示す図

【図 1 0】 本発明の実施形態 2 のプログラム変換装置による第 2 の処理を示す図

【図 1 1】 本発明の実施形態 3 の命令無効化制御方法と条件付き実行ステータス遷移図

【図 1 2】 本発明の実施形態 3 の指示命令を用いた機械語プログラム 2 を示す図

【図 1 3】 本発明の実施形態 3 のプロセッサの動作タイミング図

【図 1 4】 本発明の実施形態 3 の第 1 の書き換え前のアセンブリ言語プログ

ラムを示す図

【図 1 5】 本発明の実施形態 3 の第 2 の書き換え前のアセンブリ言語プログラムを示す図

【図 1 6】 本発明の実施形態 4 のプログラム変換装置による第 2 の処理 2 0 7 の前半部を示す図

【図 1 7】 本発明の実施形態 5 の命令無効化制御方法と条件付き実行ステータス遷移図

【図 1 8】 本発明の実施形態 5 の指示命令を用いた機械語プログラム 3 を示す図

【図 1 9】 本発明の実施形態 5 のプロセッサの動作タイミング図

【図 2 0】 本発明の実施形態 6 のプログラム変換装置の概略構成図

【図 2 1】 本発明の実施形態 6 の書き換え前のアセンブリ言語プログラムを示す図

【図 2 2】 本発明の実施形態 6 のプログラム変換装置による第 2 の処理 2 0 7 の前半部を示す図

【図 2 3】 本発明の実施形態 7 の命令無効化制御方法と条件付き実行ステータス遷移図

【図 2 4】 本発明の実施形態 7 の指示命令を用いた機械語プログラム 4 を示す図

【図 2 5】 本発明の実施形態 7 のプロセッサの動作タイミング図

【図 2 6】 本発明の実施形態 8 の書き換え前のアセンブリ言語プログラムを示す図

【図 2 7】 本発明の実施形態 8 のプログラム変換装置による第 2 の処理 2 0 7 の前半部を示す図

【図 2 8】 本発明の実施形態 9 の命令フェッチ回路 7 0 6 の構成例を示す図

【図 2 9】 本発明の本実施形態 9 の命令フェッチ回路 7 0 6 の制御方法を示す図

【図 3 0】 本発明の本実施形態 9 の命令無効化制御方法と条件付き実行ステータス 7 0 9 の遷移図

【図 3 1】 本発明の実施形態 9 のプロセッサの動作タイミング図

【図 3 2】 従来の分岐処理を示すフローチャート

【図 3 3】 従来の分岐命令によるアセンブリ言語プログラムを示す図

【図 3 4】 従来の条件付き実行命令によるアセンブリ言語プログラムを示す

図

【図 3 5】 従来の 3 ビットの条件指定フィールドの具体例を示す図

【符号の説明】

4 0 条件指定フィールド

4 1 命令コード

6 0 データ処理装置

6 1 アセンブリ言語プログラム

6 2 プログラム変換装置

6 3 機械語プログラム

6 4 プロセッサ

1 4 1 ROMからの命令

1 4 2 命令セクタ

1 4 3 命令解読の対象となる命令

2 0 1 命令列記憶装置

2 0 2 第 1 命令検出手段

2 0 3 第 2 命令検出手段

2 0 5 命令列書き換え手段

2 0 6 第 1 の処理

2 0 7 第 2 の処理

2 0 8 第 1 判定手段

2 0 9 第 2 判定手段

2 1 0 命令検索手段

2 1 1 第 2 の処理の前半部

7 0 1 排他的条件

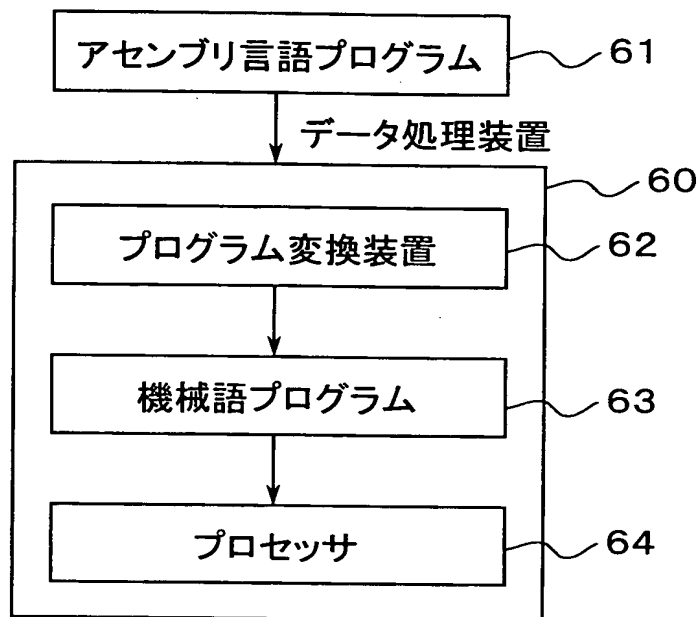
7 0 2 セクタ

- 7 0 3 演算器
- 7 0 4 レジスタファイル
- 7 0 5 命令解読器
- 7 0 6 命令フェッチ回路
- 7 0 7 命令無効化信号
- 7 0 8 命令無効化制御回路
- 7 0 9 条件付き実行ステータス
- 7 1 0 D 1 ラッチ
- 7 1 1 D 2 ラッチ
- 7 1 2 プロセッサステータスワード
- 7 1 3 命令先読み制御信号
- 7 1 4 命令ラッチ
- 7 1 5 パイプライン制御信号
- 7 1 6 シーケンサ
- 7 1 7 ROM

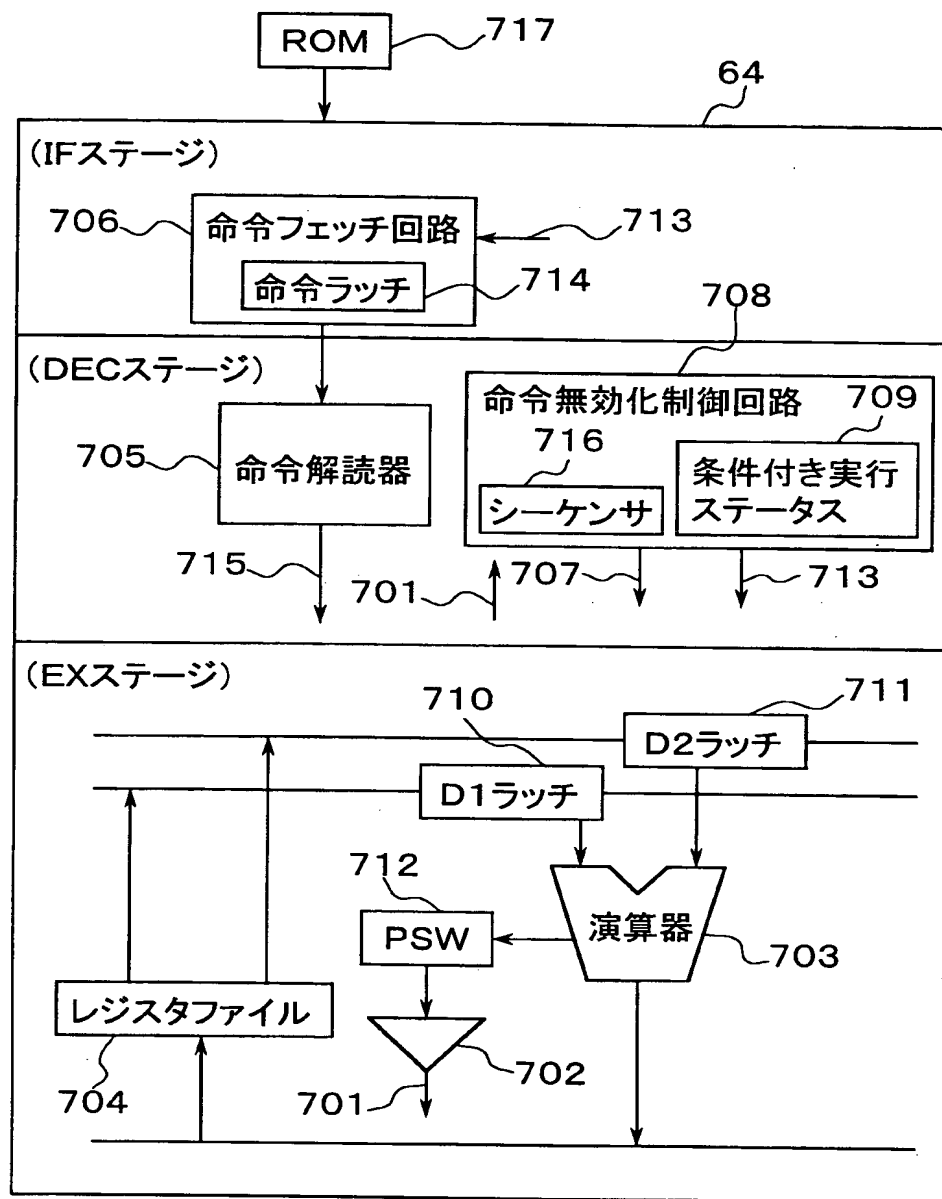


【書類名】 図面

【図 1】



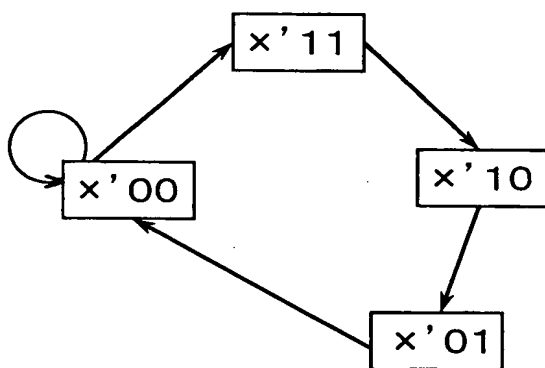
【図 2】



【図 3】

条件付き実行 ステータス 排他的条件	条件付き実行 ステータス			
	x'00	x'01	x'10	x'11
1	命令 無効化 信号 "0"	命令 無効化 信号 "1"	命令 無効化 信号 "0"	命令 無効化 信号 "0"
0	命令 無効化 信号 "0"	命令 無効化 信号 "0"	命令 無効化 信号 "1"	命令 無効化 信号 "0"

(a)



(b)

【図 4】

IF ステージ	命令 1	命令 2	命令 3	命令 4	命令 5	命令 6
	命令 1		命令 2	命令 3	命令 4	命令 5
			命令 1	命令 2	<del>命令 3</del>	命令 4

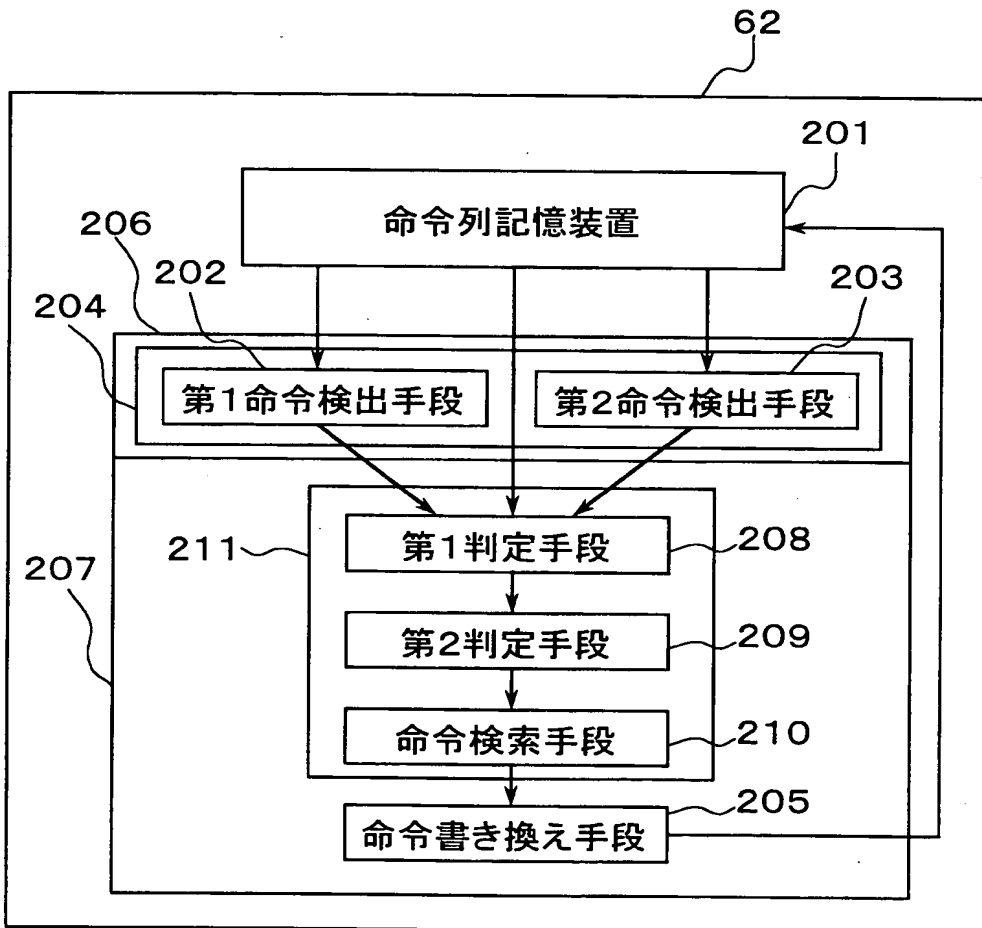
条件付き実行 ステータス	x'00	x'00	x'11	x'10	x'01	x'00	x'00
排他的条件	0	0	0	1	1	1	0
命令無効化信号	0	0	0	0	1	0	0

t1      t2      t3      t4      t5      t6  
[t11 t12] [t21 t22] [t31 t32] [t41 t42] [t51 t52] [t61 t62]

【図 5】

1行目	SUBIFEZF R0,R1
2行目	MOV R2,R4
3行目	MOV R3,R4
4行目	ADD R5,R6
5行目	NOP
6行目	NOP

【図 6】



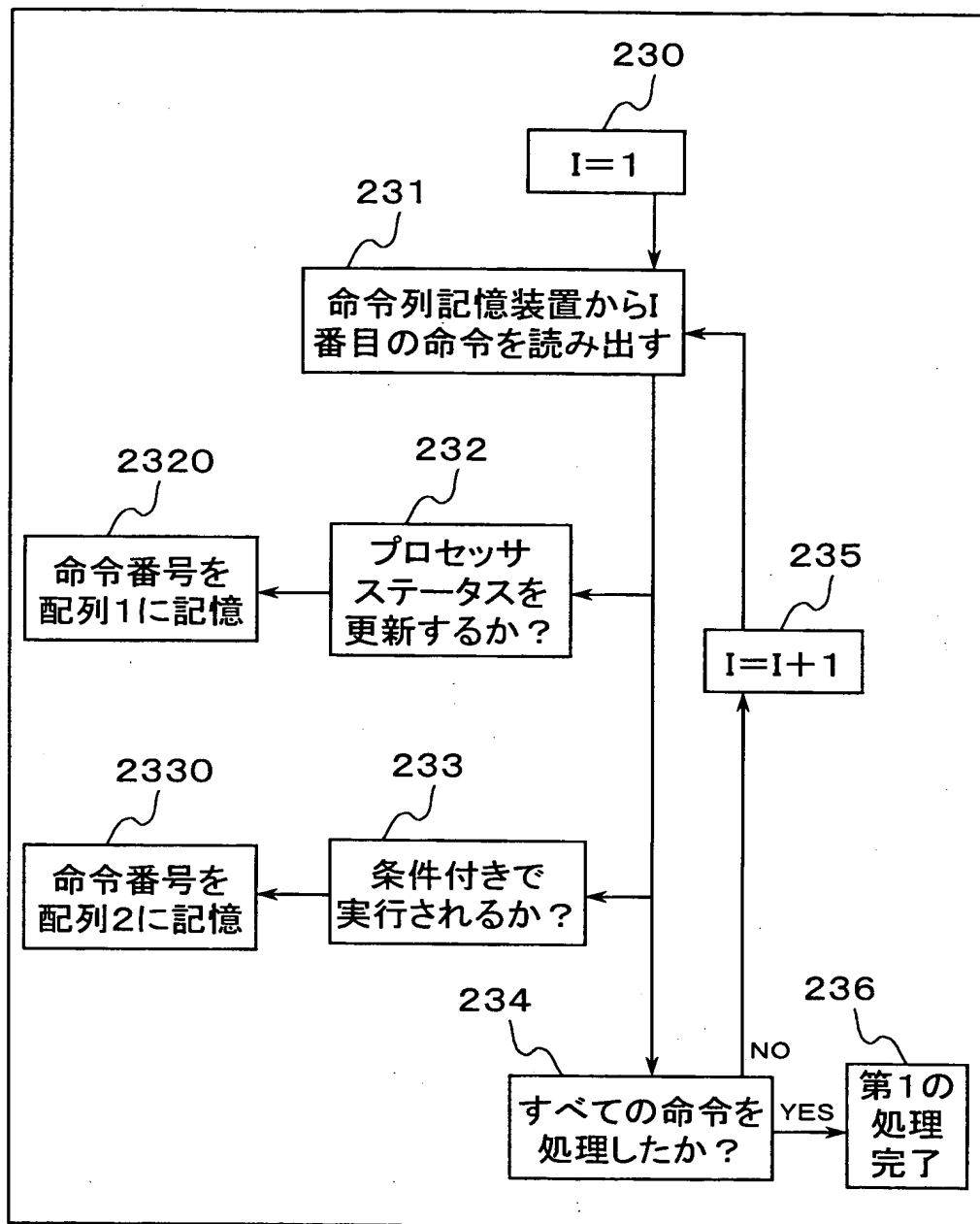
【図 7】

1行目	ADD R5,R1
2行目	SUB R0,R1
3行目	MOVIFZF R2,R4
4行目	MOVIFNZF R3,R4
5行目	ADD R4,R5
6行目	MOV R5,R7
7行目	MOVIFZF x'0001,R6
8行目	MOV R6,R0

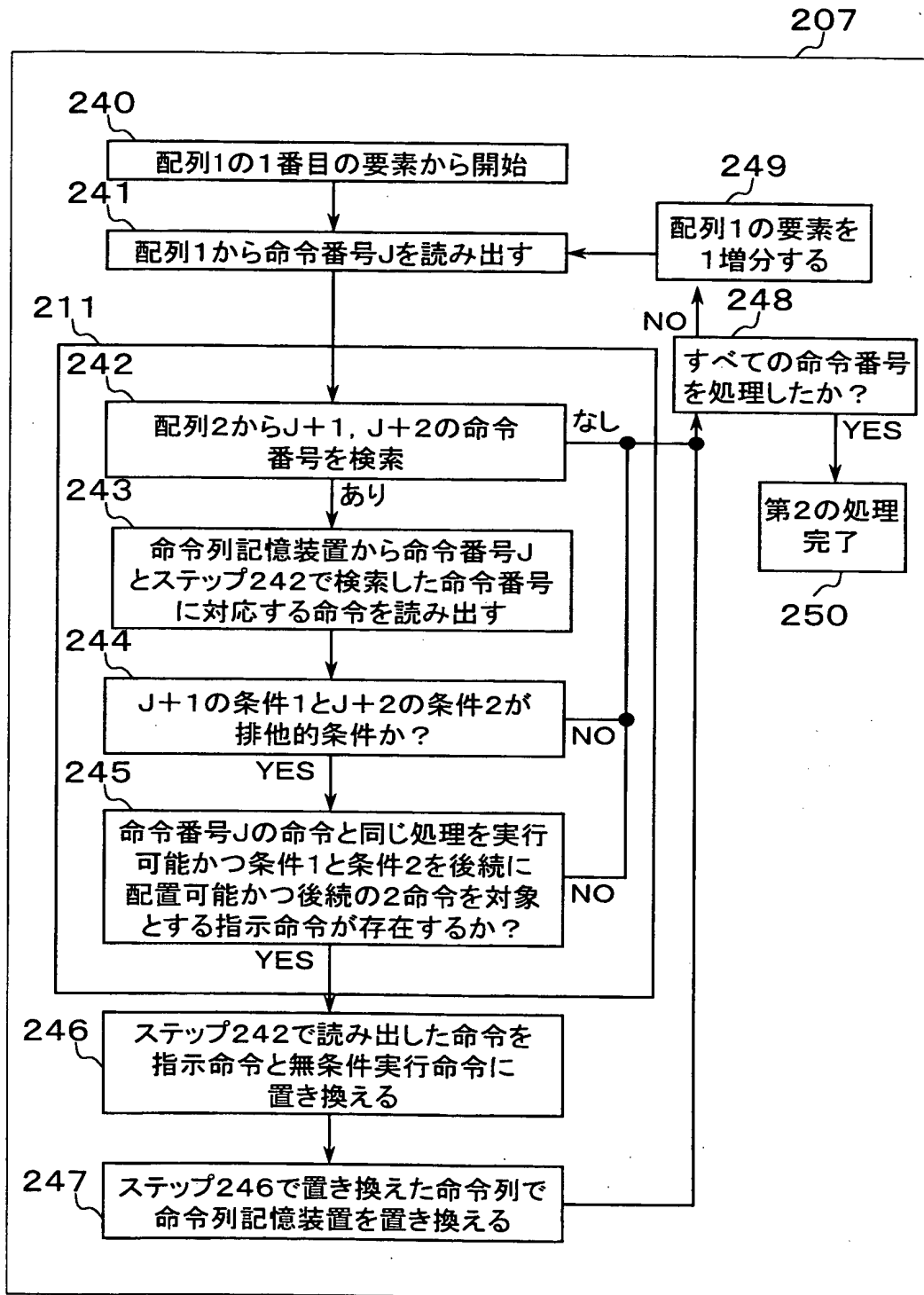
【図 8】

1行目	ADD R5,R1
2行目	SUBIFEZF R0,R1
3行目	MOV R2,R4
4行目	MOV R3,R4
5行目	ADD R4,R5
6行目	MOV R5,R7
7行目	MOVIFZF x'0001,R6
8行目	MOV R6,R0

【図 9】



【図10】

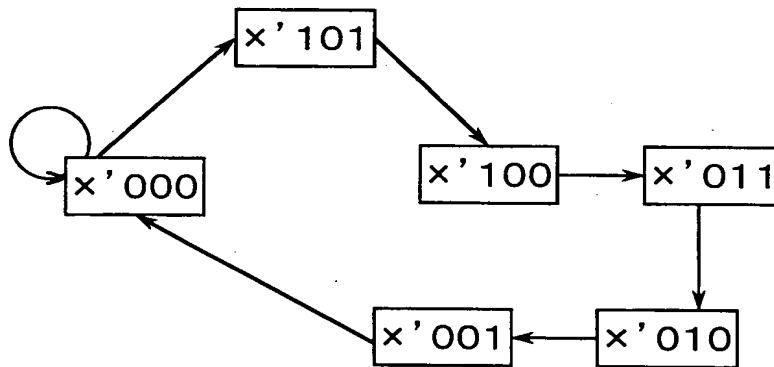




【図 1 1】

条件付き実行 ステータス 排他的条件	x'000	x'010 or x'001	x'100 or x'011	x'101
1	命令 無効化 信号 "0"	命令 無効化 信号 "1"	命令 無効化 信号 "0"	命令 無効化 信号 "0"
0	命令 無効化 信号 "0"	命令 無効化 信号 "0"	命令 無効化 信号 "1"	命令 無効化 信号 "0"

(a)



(b)

【図 1 2】

1行目	SUBIFEXZF R0,R1
2行目	MOV R2,R6
3行目	MOV R3,R7
4行目	MOV R4,R6
5行目	MOV R5,R7
6行目	MOV x'0001,R6
7行目	ADD R0,R6
8行目	NOP

【図 1 3】

IF ステージ	命令 1	命令 2	命令 3	命令 4	命令 5	命令 6	命令 7	命令 8
	命令 1		命令 2	命令 3	命令 4	命令 5	命令 6	命令 7
			命令 1	命令 2	命令 3	<del>X</del>	<del>X</del>	命令 6

条件付き実行	x'000	x'101	x'100	x'011	x'010	x'001	x'000	x'000
ステータス	0	0	1	1	1	1	1	1
排他的条件	0	0	0	0	1	1	0	0
命令無効化信号	t1	t2	t3	t4	t5	t6	t7	t8

[t11 t12] [t21 t22] [t31 t32] [t41 t42] [t51 t52] [t61 t62] [t71 t72] [t81 t82]

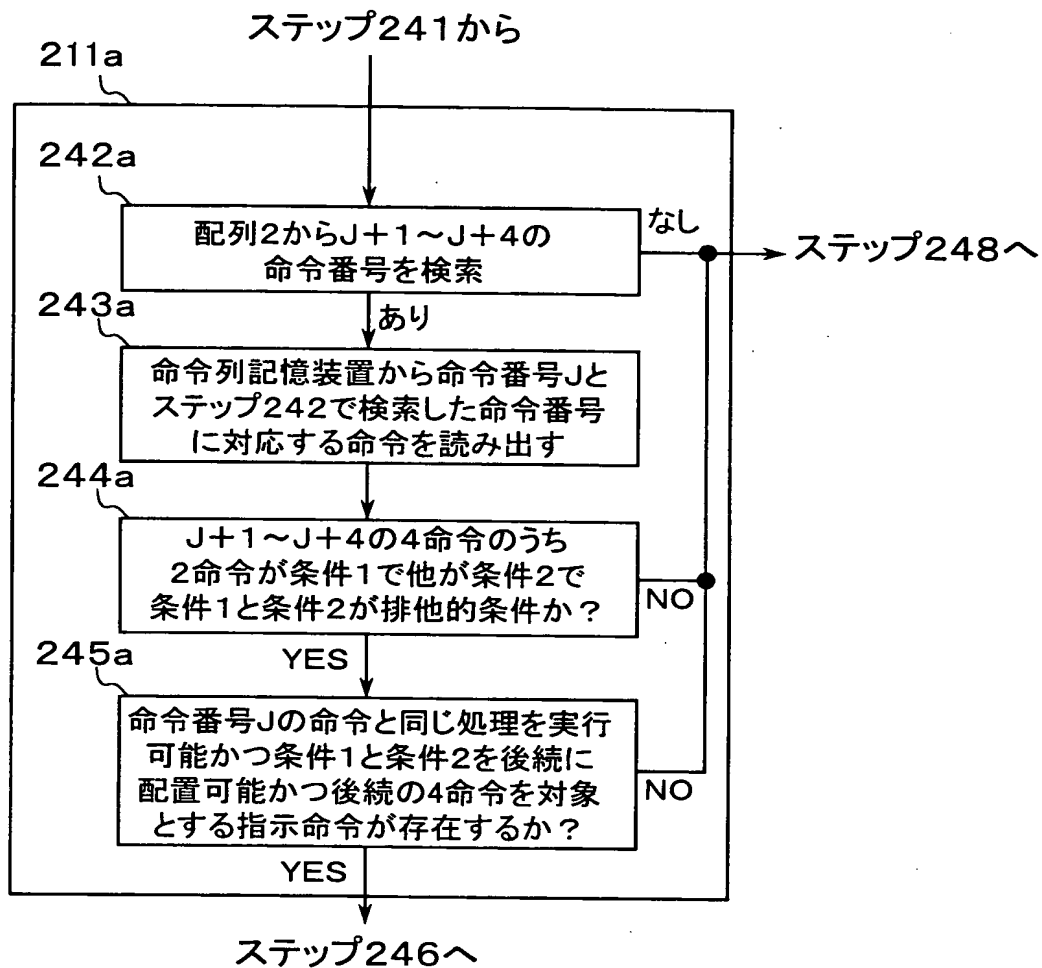
【図14】

1行目	SUB R0,R1
2行目	MOVIFZF R2,R6
3行目	MOVIFZF R3,R7
4行目	MOVIFNZF R4,R6
5行目	MOVIFNZF R5,R7
6行目	MOV x'0001,R6
7行目	ADD R0,R6
8行目	NOP

【図15】

1行目	SUB R0,R1
2行目	MOVIFZF R2,R6
3行目	MOVIFNZF R4,R6
4行目	MOVIFNZF R5,R7
5行目	MOVIFZF R3,R7
6行目	MOV x'0001,R6
7行目	ADD R0,R6
8行目	NOP

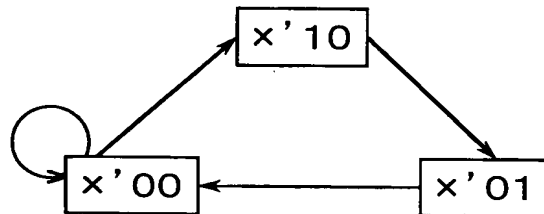
【図 16】



【図 1 7】

条件付き実行 ステータス 排他的条件	x'00	x'01	x'10
	命令 無効化 信号 "0"	命令 無効化 信号 "1"	命令 無効化 信号 "0"
1	命令 無効化 信号 "0"	命令 無効化 信号 "1"	命令 無効化 信号 "0"
0	命令 無効化 信号 "0"	命令 無効化 信号 "0"	命令 無効化 信号 "0"

(a)



(b)

【図 18】

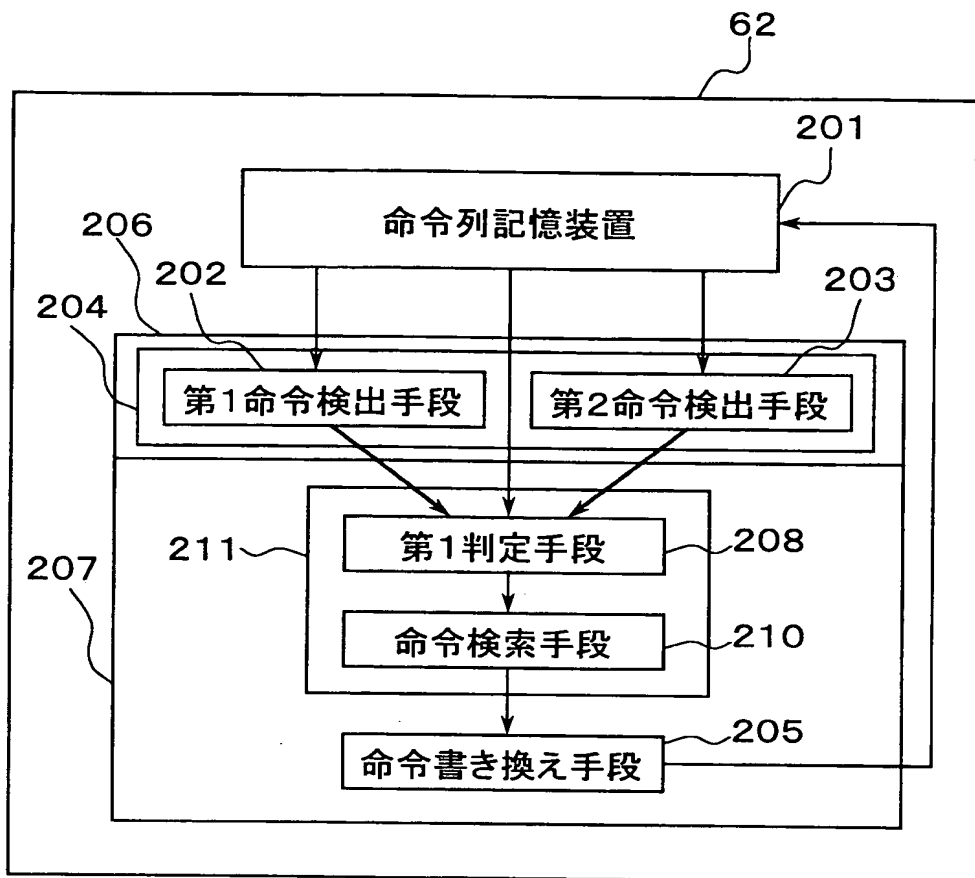
1行目	SUBIFXZF R0,R1
2行目	MOV R2,R3
3行目	ADD R3,R6
4行目	MOV R4,R7

【図 19】

IFステージ	命令1	命令2	命令3	命令4	命令5
DECステージ		命令1	命令2	命令3	命令4
EXステージ			命令1	<del>命令3</del>	命令3

条件付き実行 ステータス	x'00	x'00	x'10	x'01	x'00	x'00
排他的条件	0	0	0	1	1	1
命令無効化信号	0	0	0	1	0	0
	t1	t2	t3	t4	t5	
	[t11 t12]	[t21 t22]	[t31 t32]	[t41 t42]	[t51 t52]	

【図 20】

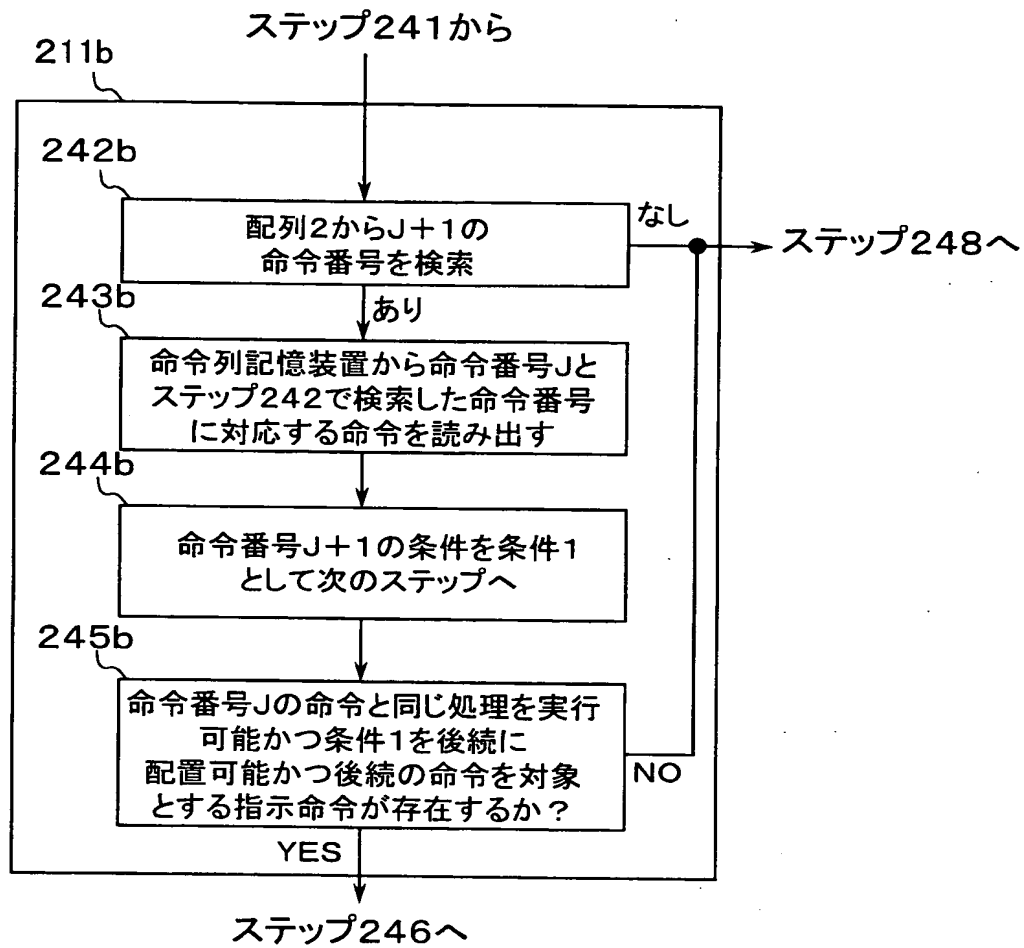


【図 21】

1行目	SUB R0,R1
2行目	MOVIFNZF R2,R3
3行目	ADD R3,R6
4行目	MOV R4,R7



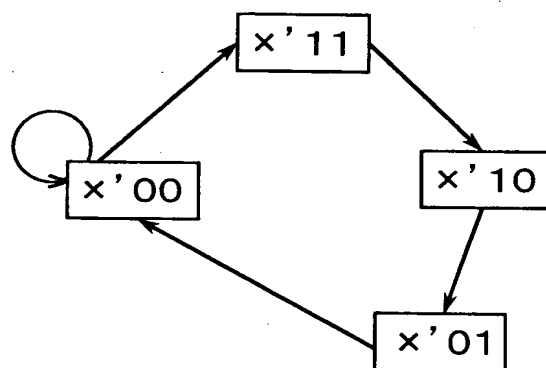
【図 22】



【図 2 3】

条件付き実行 ステータス 排他的条件	x'00	x'01	x'10	x'11
1	命令 無効化 信号 "0"	命令 無効化 信号 "1"	命令 無効化 信号 "1"	命令 無効化 信号 "0"
0	命令 無効化 信号 "0"	命令 無効化 信号 "0"	命令 無効化 信号 "0"	命令 無効化 信号 "0"

(a)



(b)

【図 2 4】

1行目	SUBIFNXZF R0,R1
2行目	MOV R2,R3
3行目	MOV R5,R4
4行目	ADD R3,R6
5行目	MOV R4,R7

【図 25】

IFステージ	命令1	命令2	命令3	命令4	命令5	命令6
	命令1		命令2	命令3	命令4	命令5
			命令1	<del>命令3</del>	<del>命令4</del>	命令4

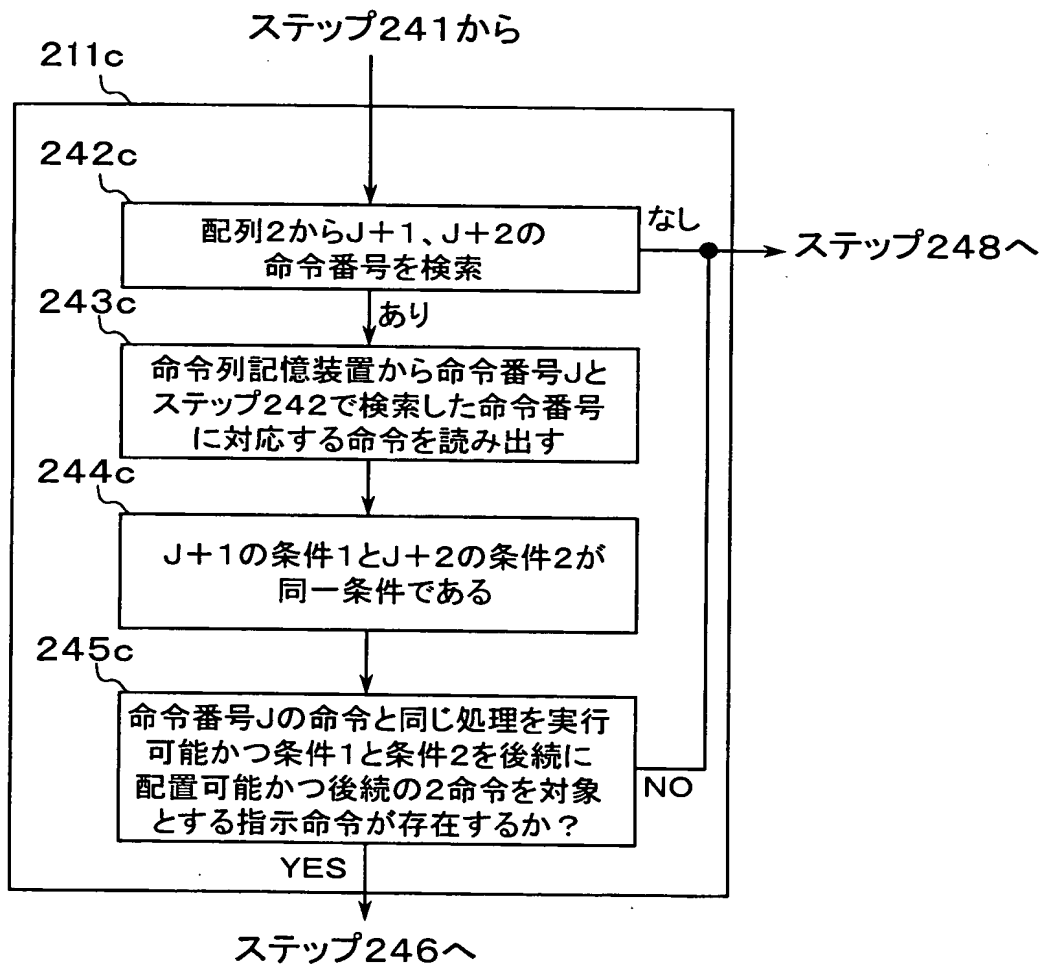
条件付き実行 ステータス	x'00	x'00	x'11	x'10	x'01	x'00	x'00
排他的条件	0	0	0	1	1	1	1
命令無効化信号	0	0	0	1	1	0	0

$t1$     $t2$     $t3$     $t4$     $t5$     $t6$   
 $[t11 \ t12] \ [t21 \ t22] \ [t31 \ t32] \ [t41 \ t42] \ [t51 \ t52] \ [t61 \ t62]$

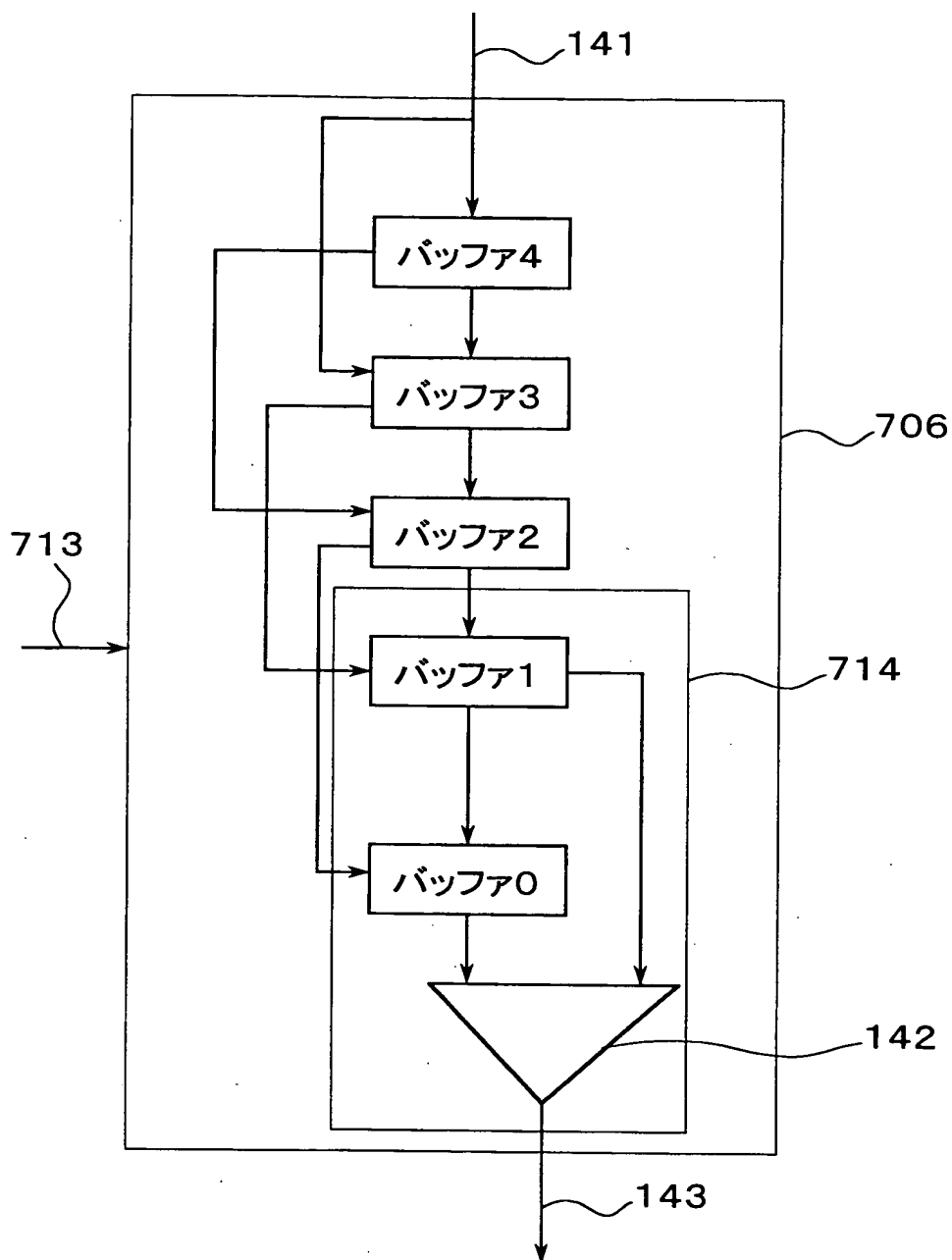
【図 2 6】

1行目	SUB R0,R1
2行目	MOVIFNZF R2,R3
3行目	MOVIFNZF R5,R4
4行目	ADD R3,R6
5行目	MOV R4,R7

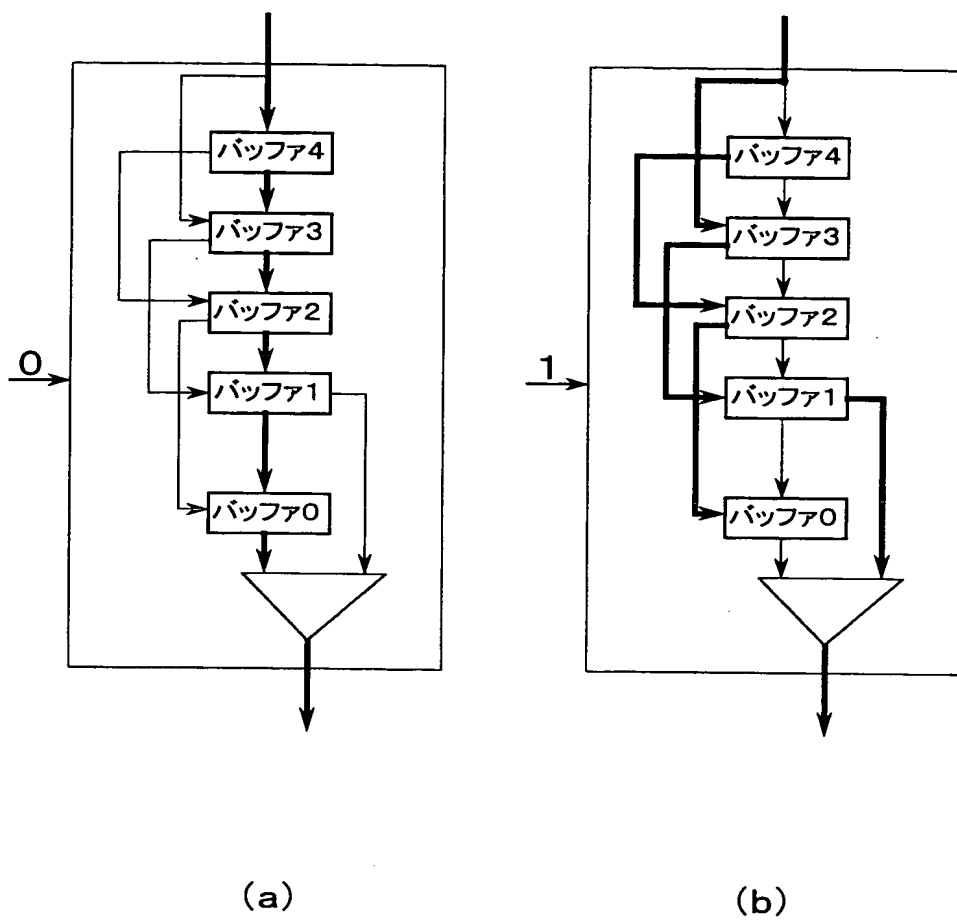
【図 2 7】



【図 28】



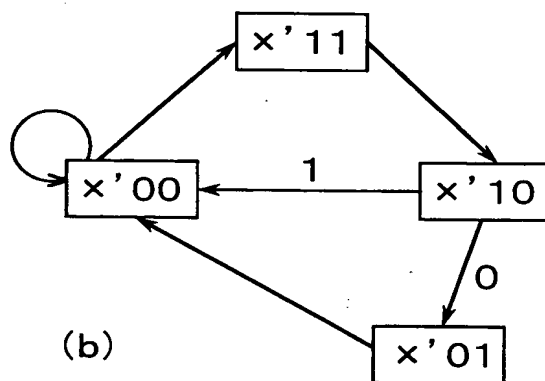
【図 2 9】



【図 30】

条件付き実行 ステータス 排他的条件	x'00	x'01	x'10	x'11
1	命令 無効化 信号 "0"	×	命令 無効化 信号 "0"	命令 無効化 信号 "0"
0	命令 無効化 信号 "0"	命令 無効化 信号 "0"	命令 無効化 信号 "1"	命令 無効化 信号 "0"

(a)



(b)

条件付き実行 ステータス 排他的条件	x'00	x'01	x'10	x'11
1	0	0	1	0
0	0	0	0	0

(c)



【図 3 1】

バッファ4	命令5	命令6	命令7	—	—
バッファ3	命令4	命令5	命令6	命令8	命令9
バッファ2	命令3	命令4	命令5	命令7	命令8
バッファ1	命令2	命令3	命令4	命令6	命令7
バッファ0	命令1	命令2	命令3	命令5	命令6
IFステージ	命令1	命令2	命令3	命令5	命令6
DECステージ	命令1		命令2	命令4	命令5
EXステージ			命令1	命令2	命令4

条件付き実行 ステータス	x'00	x'00	x'11	x'10	x'00	x'00
命令先読み制御信号	0	0	0	1	0	0
排他的条件	0	0	0	1	1	1
命令無効化信号	0	0	0	0	0	0

t1

[t11 t12]

t2

[t21 t22]

t3

[t31 t32]

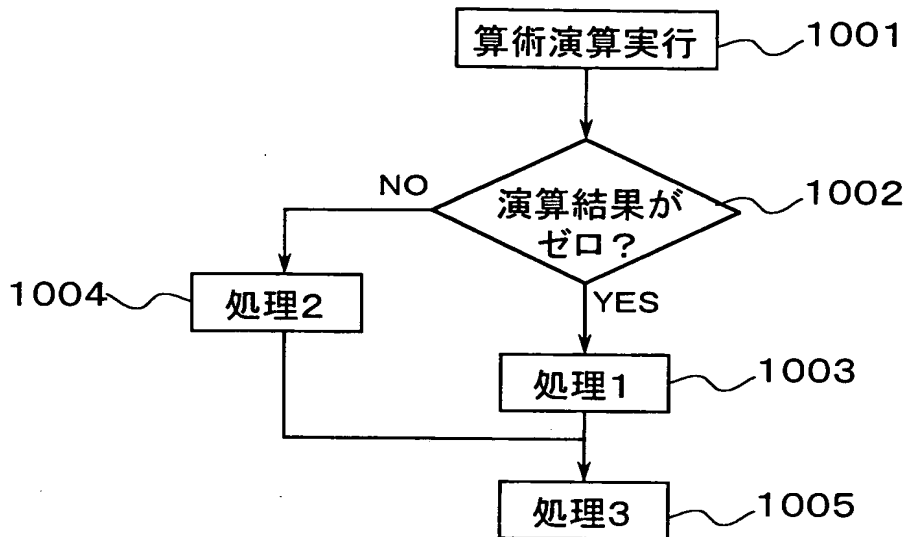
t4

[t41 t42]

t5

[t51 t52]

【図 3 2】



【図 3 3】

1行目	SUB R0,R1
2行目	BNE label1
3行目	命令1
4行目	JMP label2
5行目	label1: 命令2
6行目	label2: 命令3

【図 3 4】

1行目	SUB R0,R1
2行目	MOVIFZF R2,R4
3行目	MOVIFNZF R3,R4
4行目	ADD R5,R6

【図 35】

条件指定フィールド	命令コード
000	キャリーフラグが1
001	キャリーフラグが0
010	ネガティブフラグが1
011	ネガティブフラグが0
100	オーバーフローフラグが1
101	オーバーフローフラグが0
110	ゼロフラグが1
111	ゼロフラグが0

【書類名】 要約書

【要約】

【課題】 条件付き実行命令の実行にあたり、条件指定フィールドを付加した場合の命令割り付けやメモリ少量化の弊害を防止し、相反する条件付き命令が連続した場合でも条件不成立によるパイプラインハザードの発生を抑制するデータ処理装置を提供する。

【解決手段】 パイプライン処理するデータ処理装置であって、実行するための条件が明示されない命令 1 と、その直後に配置された同じく実行するための条件が明示されない命令 2 と、命令 1 の直前に配置され命令 1 と命令 2 を実行するための条件が互いに相反することを明示した指示命令と、指示命令が実行され条件が確定すると同時に命令 1 と命令 2 のどちらか一方を無効化する条件付き実行制御手段を備える。さらに後方に配置された命令が無効化されるとき命令先読み手段から無効化すべき命令をスキップして命令解読手段に命令を転送させる。

【選択図】 図 2

出 願 人 履 歴 情 報

識別番号 [000005821]

1. 変更年月日 1990年 8月28日  
[変更理由] 新規登録  
住 所 大阪府門真市大字門真1006番地  
氏 名 松下電器産業株式会社